BW-CAR | SINCOM
SYMPOSIUM ON INFORMATION
AND COMMUNICATION SYSTEMS

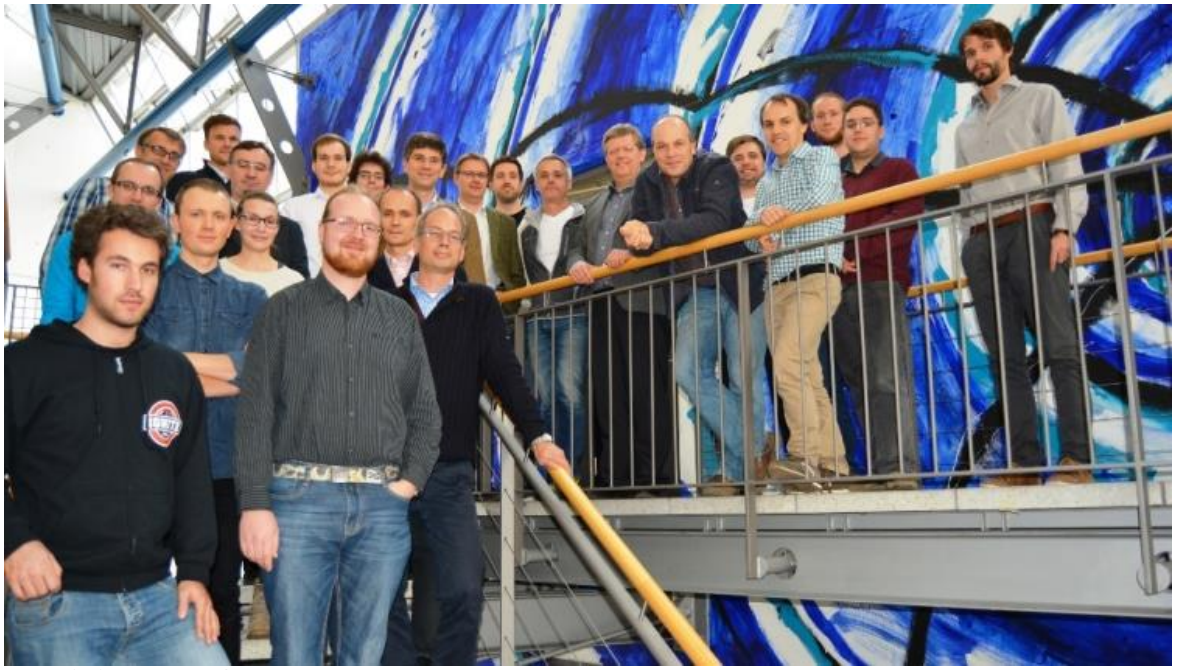
2nd Baden-Württemberg Center of Applied Research
Symposium on
Information and Communication Systems

SInCom 2015

13. November 2015
in Konstanz

ISBN 978-3-00-051859-1

HOCHSCHULE
KONSTANZ
UNIVERSITY OF
APPLIED SCIENCES



Participants of SInCom 2015

Message from the ProgramChairs

The Baden-Württemberg Center of Applied Research (BW-CAR) intends to further develop the applied research at the Universities of Applied Science (UAS). The BW-CAR working group "Informations- und Kommunikationssysteme" (IKS) in cooperation with the working group "Technologien für Intelligente Systeme" organized the 2nd BW-CAR Symposium on Information and Communication Systems (SInCom) at the HTWG Konstanz, university of applied science. The IKS working group represents various disciplines and application areas with expertise for different aspects of communication technologies, communication methods and procedures for processing in information systems. The collection, analysis, processing, transfer and output of information are the key technologies of information and communication systems, which in turn are inherent components of any modern technical system. These key technologies are found in factories, mobile networks, smart grids, navigation systems, ambient assisted living, environmental engineering, in macroscopic or in embedded systems such as smart phones, and sensor networks.

The SInCom 2015 aimed at young researchers for contribution in the fields of

- Distributed Computing
- Communication Networks
- Security Systems
- Algorithms, Signal and Image Processing

Many thanks to all authors for the valuable contributions to SInCom 2015. Furthermore, we would like to thank the reviewers for their suggestions for improvement. Without them no high-quality conference proceedings could have been achieved.

Konstanz, 10.12.2015
Prof. Dr. Dirk Benyoucef and Prof. Dr. Jürgen Freudenberger

Organizing Committee

Prof. Dr. Dirk Benyoucef, Hochschule Furtwangen

Prof. Dr. Jürgen Freudenberger, Hochschule Konstanz

Program Committee

Prof. Dr. Dirk Benyoucef, Hochschule Furtwangen
Prof. Dr.-Ing. Andreas Christ, Hochschule Offenburg
Prof. Dr. rer. nat. Thomas Eppler, Hochschule Albstadt-Sigmaringen
Prof. Dr. Matthias Franz, Hochschule Konstanz
Prof. Dr.-Ing. Jürgen Freudenberger, Hochschule Konstanz
Prof. Dr. Thomas Greiner, Hochschule Pforzheim
Prof. Dr. rer.nat. Roland Münzer, Hochschule Ulm
Prof. Dr.-Ing. Franz Quin, Hochschule Karlsruhe
Prof. Dr. Christoph Reich, Hochschule Furtwangen
Prof. Dr. Georg Umlauf, Hochschule Konstanz
Prof. Dr.-Ing. Axel Sikora, Hochschule Offenburg
Prof. Dr. Peter Väterlein, Hochschule Esslingen
Prof.Dr. Dirk Westhoff, Hochschule Offenburg

Content

Soft Input Decoding of Generalized Concatenated Codes Using a Stack Decoding Algorithm

Jens Spinner and Jürgen Freudenberger
HTWG Konstanz
University of Applied Sciences, Konstanz, Germany
Institute for System Dynamics (ISD)
Email: {jens.spinner,juergen.freudenberger}@htwg-konstanz.de

***Abstract*—This work investigates soft input decoding for generalized concatenated (GC) codes. The GC codes are constructed from inner nested binary Bose-Chaudhuri-Hocquenghem (BCH) codes and outer Reed-Solomon (RS) codes. In order to enable soft input decoding for the inner BCH block codes, a sequential stack decoding algorithm is used. Ordinary stack decoding of binary block codes requires the complete trellis of the code. In this work a representation of the block codes based on the trellises of supercodes is proposed in order to reduce the memory requirements for the representation of the BCH codes. Results for the decoding performance of the overall GC code are presented. Furthermore, an efficient hardware implementation of the GC decoder is proposed.**

I. Introduction

Error correction coding (ECC) based on GC codes has a high potential for various applications in data communication and data storage systems, e.g., for digital magnetic storage systems [1], for non-volatile flash memories [2], and for two-dimensional bar codes [3]. GC codes have a low decoding complexity compared to long BCH codes [4]. The residual error rates for GC codes can be determined analytically [5], which is important for industry applications where a low probability of a system failure has to be guaranteed. In [4] a pipelined decoder architecture for GC codes was proposed which is based on algebraic hard input decoding of the component codes. In this work we extend this design to soft input decoding.

A codeword of a GC code can be considered as a matrix. For encoding the information is stored in the matrix. In the first encoding step the rows of the matrix are protected by block codes (the outer codes) over the Galois field $GF(2^m)$. Next each column is protected with binary codes, the inner codes. Typically binary BCH codes are used as inner codes and RS codes as outer codes [6], [7]. A decoder processes the erroneous data in multiple decoding steps. In [4] algebraic decoding is used in each decoding step. This is adequate if the channel provides no soft information about the transmitted or stored bits, e.g., for the binary symmetric channel (BSC). However, if the channel provides reliability information, e.g. an additive white Gaussian noise (AWGN) channel, this soft information should be exploited by the decoder. In the case of GC code, it is sufficient to decode the inner codes exploiting the soft information.

There exist numerous soft input decoding algorithms for binary block codes [6]. However, many of these methods would not be suitable for a fast hardware implementation. In this work we consider sequential stack decoding as proposed in [8]. Sequential decoding has a low computational complexity, if the noise level is small. This is the case for many applications of GC codes, e.g., for error correction coding in storage systems.

Sequential decoding was original introduced for tree codes. In order to decode binary block codes, the syndrome trellis is used as a representation of the code [9]. For block codes the number of trellis states grows exponentially with the number of redundancy bits. Hence, the trellis based sequential decoding as proposed in [8] is only feasible for codes with low error correcting capabilities. In this work, we use this algorithm only in the first decoding iteration for the inner BCH codes. The GC construction is based on nested-BCH codes where higher levels have higher error correcting capabilities. The partitioning of the nested-BCH codes can be used to reduce the space complexity required for representing the code.

We propose a representation based on supercodes. A similar method was introduced in [10] to reduce the decoding complexity of maximum-likelihood decoding. A supercode is a superset $\mathcal{D}_1$ of the original code $\mathcal{D} \subset \mathcal{D}_1$. In order to decode the original code $\mathcal{D}$, two supercodes $\mathcal{D}_1$ and $\mathcal{D}_2$ have to be constructed such that $\mathcal{D}_1 \cap \mathcal{D}_2 = \mathcal{D}$. The supercodes have fewer redundancy bits and thus fewer trellis states. The supercodes can be constructed such that each code has half of the original redundancy bits. This reduces the number of states from $O(2^r)$ to $O(2^{\frac{r}{2}})$ in standard order notation.

In this work we give a brief introduction into the GC construction, explaining its structure and the decoding algorithm. Later on we describe the stack algorithm. This algorithm is used to decode the nested-BCH codes. In section IV we describe the proposed supercode decoding algorithm. Next an implementation of the soft decoding of the GC codes is presented. Finally we show results for the error correction performance and decoding complexity.

II. GC Construction

In this section we explain the GC construction and its parameters. A detailed discussion can be found in [5]. The GC codeword is arranged in an $n_b \times n_a$ matrix as depicted

in Fig. 1, where each column is a codeword of a binary BCH code $\mathcal{B}$ of length n_b. The rows are protected by RS codes of length n_a. m elements of each column represent one symbol $a_{j,i}$ from the Galois field $GF(2^m)$. Hence, m rows form a codeword of an outer code $\mathcal{A}^{(i)}$.

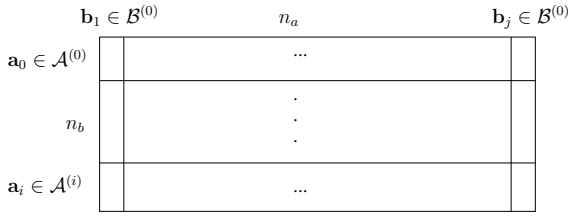


Fig. 1. GC data matrix

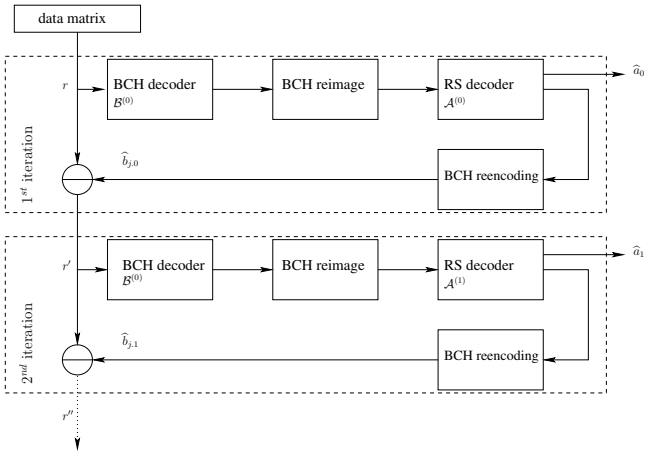


Fig. 2. GC decoding schemes

Each column is the sum of L codewords of nested linear BCH codes.

$$\mathcal{B}^{(L-1)} \subset \mathcal{B}^{(L-2)} \subset \ldots \subset \mathcal{B}^{(0)} \tag{1}$$

Hence, a higher level code is a sub-code of its predecessor, where the higher levels have higher error correcting capabilities, i.e., $t_{b,L-1} \geq t_{b,L-2} \geq \ldots \geq, t_{b,0}$, where $t_{b,i}$ is the error correcting capability of level i. The codeword of the j-th column is the sum of L codewords. These codewords $\mathbf{b}_j^{(i)}$ are formed by encoding the symbols $a_{j,i}$ for $0 \leq i < L$ with the corresponding sub-code $\mathcal{B}^{(i)}$. For this encoding $(L-i-1)m$ zero bits are prefixed onto the symbol $a_{j,i}$. The final column codeword is

$$\mathbf{b}_j = \sum_{i=0}^{L-1} \mathbf{b}_j^{(i)}. \tag{2}$$

Because of the linearity of the nested codes, $\mathbf{b}_j$ is a codeword of $\mathcal{B}^{(0)}$.

The decoder processes level by level starting with $i = 0$. Fig. 2 depicts the decoding steps. Let i be the index of the current level. First the columns are decoded with respect to $\mathcal{B}^{(i)}$ and the information bits have to be inferred (re-image) in order to retrieve the code symbols $a_{i,j}$ of $\mathcal{A}^{(i)}$ where j the column index. If all symbols of the code $\mathcal{A}^{(i)}$ are inferred the RS code can be decoded. At this point a partial decoding result $\hat{a_i}$ is available. Finally this result has to be re-encoded using $\mathcal{B}^{(i)}$. The estimated codewords of the inner code $\mathcal{B}^{(i)}$ are subtracted from the codeword matrix $\mathcal{C}$ before the next level can be decoded. The detailed encoding and decoding process is described in [4].

III. Stack Algorithm

In this section we describe the sequential decoding procedure using the stack algorithm for block codes as presented in [8]. This procedure uses the trellis representation. A trellis $\mathcal{T} = (\mathcal{S}, \mathcal{W})$ is a labeled, directed graph, where $\mathcal{W} = \{w\}$ denotes the set of all branches in the graph and $\mathcal{S} = \{\sigma\}$ is the set of all nodes. The set $\mathcal{S}$ is decomposed into $n+1$ disjoint subsets $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_n$ that are called levels of the trellis. Similarly, there exists a partition of the set $\mathcal{W} = \mathcal{W}_1 \cup \mathcal{W}_2 \cup \ldots \cup \mathcal{W}_n$. A node $\sigma \in \mathcal{S}_t$ of the level t may be connected with a node $\widetilde{\sigma} \in \mathcal{S}_{t+1}$ of the level $t+1$ by one or several branches. Each branch w_t is directed from a node σ of level $t-1$ to a node $\widetilde{\sigma}$ of the next level t. We assume that the end levels have only one node, namely $\mathcal{S}_0 = \{\sigma_0\}$ and $\mathcal{S}_n = \{\sigma_n\}$. A trellis is a compact method of presenting all codewords of a code. Each branch of the trellis w_t is labeled by a code symbol $\mathrm{v}_t(w_t)$. Each distinct codeword corresponds to a distinct path in the trellis, i.e., there is a one-to-one correspondence between each codeword $\mathbf{v}$ in the code and a path $\mathbf{w}$ in the trellis: $\mathbf{v}(\mathbf{w}) = \mathrm{v}_1(w_1), \ldots, \mathrm{v}_n(w_n)$. We denote code sequence segments and path segments by $\mathbf{v}_{[i,j]} = \mathrm{v}_i, \ldots, \mathrm{v}_j$ and $\mathbf{w}_{[i,j]} = w_i, \ldots, w_j$, respectively. The *syndrome trellis*, can be obtained using its parity-check matrix [9]. The syndrome trellis is minimal inasmuch as this trellis has the minimal possible number of nodes $\mid \mathcal{S} \mid$ among all possible trellis representations of the same code. With the syndrome trellis we also introduce a node labeling. The nodes of the trellis will be identified by $(n-k)$-tuples with elements from $\mathbb{F}_2$, with $\mathbf{0}$ referring to the all zero $(n-k)$-tuple. At level $t = 0$ and level $t = n$ the trellis contains only one node, the all zero node $\sigma_0 = \sigma_n = \mathbf{0}$.

The sequential decoding procedure as presented in [8] is a stack algorithm, i.e., a stack is required to store interim results. The stack contains code sequences of different lengths. Let $\mathbf{v}_t$ denote a code sequence of length t, i.e.$\mathbf{v}_t = \mathrm{v}_1, \ldots, \mathrm{v}_t$. Each code sequence is associated with a metric and a node σ_t. The node s_t is the node in the trellis that is reached if we follow the path corresponding to the code sequence through the trellis. The metric rates each code sequence and the stack is ordered according to these metric values where the code sequence at the top of the stack is the one with the largest metric value. There exits different metrics in the literature to compare code sequences of different length. In the following, we consider the Fano metric which is defined as follows. Let v_i be the i-th code bit and r_i the i-th received symbol for transmission over a discrete memoryless channel. The Fano metric for a code

bit v_i is defined by

$$M(r_i|v_i) = \log_2 \frac{p(r_i|v_i)}{p(r_i)} - R \tag{3}$$

where $p(r_i|v_i)$ is the channel transition probability, $p(r_i)$ is the probability to observe r_i at the channel output, and R is the code rate. The Fano metric of a code sequence $\mathbf{v}_t$ is

$$M(\mathbf{r}_t|\mathbf{v}_t) = \sum_{i=1}^{t} M(r_i|v_i) \tag{4}$$

where $\mathbf{r}_t$ is the sequence of the first t received symbols.

IV. Proposed Supercode Decoding for nested-BCH Code

In this section we first describe the supercode decoding [11]. Then we discuss the proposed application of supercode decoding for the nested-BCH codes that are used in the GC code.

A supercode $\mathcal{D}_i$ of a block code $\mathcal{D}$ is a code containing all codewords of $\mathcal{D}$. For a linear code $\mathcal{D}$ we can construct two supercodes $\mathcal{D}_1$ and $\mathcal{D}_2$ such that $\mathcal{D} = \mathcal{D}_1 \cap \mathcal{D}_2$. Let $\mathbf{H} = \begin{pmatrix} \tilde{\mathbf{H}}_1 \\ \tilde{\mathbf{H}}_2 \end{pmatrix}$ be the parity-check matrix of the code $\mathcal{D}$, this means that $\tilde{\mathbf{H}}_1$ and $\tilde{\mathbf{H}}_2$ are two sub-matrices of $\mathbf{H}$. The sub-matrices $\tilde{\mathbf{H}}_1$ and $\tilde{\mathbf{H}}_2$ define the supercodes $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively.

As already described the higher level of the code $\mathcal{B}^{(i)}$ is a subcode of its predecessor $\mathcal{B}^{(i-1)}$. We can now form distinct supercodes $\tilde{\mathcal{B}}^{(i)}$ for each level of $\mathcal{B}^{(i)}$ with $i = 1, ..., L-1$. We can form the parity-check matrices of level L by

$$\mathbf{H} = \begin{pmatrix} \tilde{\mathbf{H}}_1 \\ \vdots \\ \tilde{\mathbf{H}}_L \end{pmatrix} \tag{5}$$

where $\tilde{\mathbf{H}}_i$ is the corresponding parity-check matrix of the supercode $\tilde{\mathcal{B}}^{(i)}$.

Next we state the proposed sequential decoding algorithm for two supercodes. Any path stored in the stack is associated with a metric value as well as two states $\sigma_{t,1}$ and $\sigma_{t,2}$ which are the states in the trellis for supercode $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively.

Algorithm 1: The sequential decoding starts in the nodes $\sigma_{0,1}$ and $\sigma_{0,2}$ of the supercode trellises. Calculate the metric values for $v_1 = 0$ and $v_1 = 1$. Insert both paths into the stack according to their metric values. In each iteration, remove the code sequence at the top from the stack. Verify whether the branches for $v_{t+1} = 0$ and $v_{t+1} = 1$ exist for both nodes $\sigma_{t,1}$ and $\sigma_{t,2}$ corresponding to the top path. If a branch exists in both supercode trellises then calculate the metric for this path and insert the code sequence into the stack. The algorithm terminates when a path approaches the end nodes $\sigma_{n,1}$ and $\sigma_{n,2}$. The estimated codeword is the top path in the final iteration.

V. Decoder architecture

In this section we discuss the decoder architecture of a GC decoder. First we discuss the integration of the stack algorithm as inner decoder into the implementation of the GC decoder presented in [4]. Then the stack algorithm implementation for supercode decoding with its subsystems is presented and discussed.

The original hard input GC decoder implementation in [4] uses algebraic syndrome decoding. In this implementation the first levels of $\mathcal{B}$ can decode $t_{b,0} = 1$ and $t_{b,1} = 2$ errors. Thus high error correction capabilities of the outer codes $\mathcal{A}^{(0)}$ and $\mathcal{A}^{(1)}$ are required. This leads to lower code rates and a high decoding complexity of those outer codes. On the other hand the soft decoding complexity of the column codes increases significantly with each code level. Hence soft decoding is of interest for the lower levels.

Subsequently the algebraic decoding logic for the column code remains in the implementation. Therefore it is possible to check whether the syndrome is zero. In this case the codeword can be assumed to be correct, i.e., neither algebraic decoding nor sequential decoding result in a different codeword.

A. System overview

A brief system overview is depicted in Fig. 3. The system consists of a word array of size n_b and a desired width which stores the q-ary word. Furthermore a demultiplexer selects the currently processes bit position depending on the top path of the stack and delivers this value to the metric calculator. Based on the received codeword symbol r_i and the previous metric $M(r_{t-1}|v_{t-1})$ the metric module returns $M(r_t|v_t)$ to the priority queue block. To represent the supercode trellis asynchronous ROM is used. Each word of the ROM represents a trellis node $\sigma_{t,i}$. The data consists of two pointers for the successor nodes $v_{t+1} = 0$ and $v_{t+1} = 1$.

Depending on the top entry of the priority queue the desired codeword symbol is selected and the next branches for the actual nodes $\sigma_{t,1}$ and $\sigma_{t,2}$ are loaded from the trellis ROM. The priority queue unloads the top entry and loads the new paths in a single clock cycle.

B. Priority queue

Each entry of the priority queue contains several elements. The first element is the metric value. The path in the trellis, the length of the path, and a pointer to the current node are stored. All entries have to be ordered by the metric values such that the top entry has the highest value.

The process of the priority queue starts with its initialization. The starting node, its initial metric value and the path length are set. Each update cycle begins with the load phase in which the next node pointers are loaded from the trellis ROM. Simultaneously the next codeword symbol is loaded based on the path length index. The next metric value can be determined based on the code symbol and the available branches.

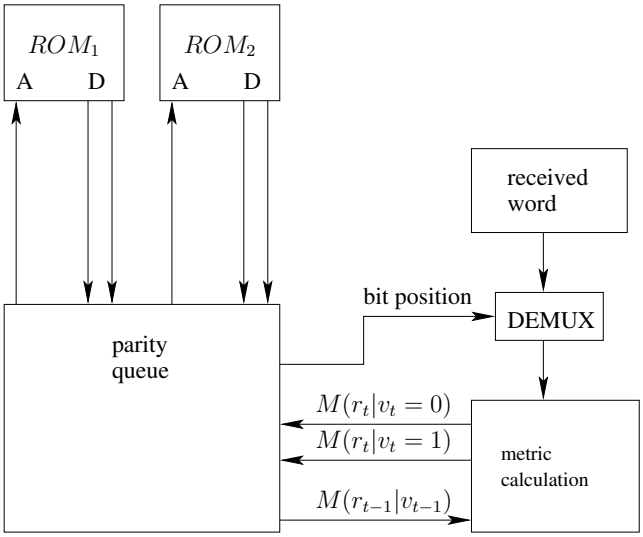


Fig. 3. Block diagram of the sequential decoder

With binary codes there exists at least one possible branch and at most two branches. The resulting branches are presorted using combinatorial logic. In the following we call these two entries the major and the minor entries, where the major entry has the better metric value.

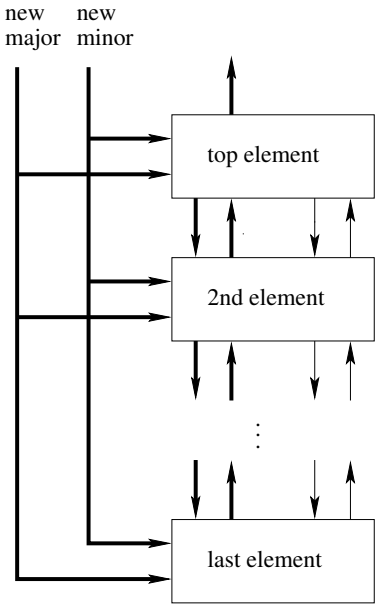


Fig. 4. Priority Queue

As depicted in Fig. 4, all priority queue elements are successively ordered in a chain. Each element can exchange its date with its previous or next neighbor. Furthermore each element can decide weather it keeps its own data, take the data from its neighbor, load the new major data or the new minor data. In each element the metric value is compared with the new value. The result of this comparison is signaled to its predecessor and successor elements. If the signal of a predecessor is false and the major metric value comparator gives a positive signal the new major value will be stored. Likewise if an element receives a false signal from its successor and the minor metric value comparator signals a new metric value that is less than the current value, the new minor data is stored. In the case that an element receives a signal from its neighbors, space for the new data has to be created by shifting all entries to next element.

There exists two special cases that have to be taken into account. The first special case occurs if a node has only a single outgoing branch. In this case the shifting of elements has to be prevented by signaling. The second special case occurs if the new major and the new minor elements are designated to be inserted into the same entry register. This case can be detected and preventing by passing this value to the next element.

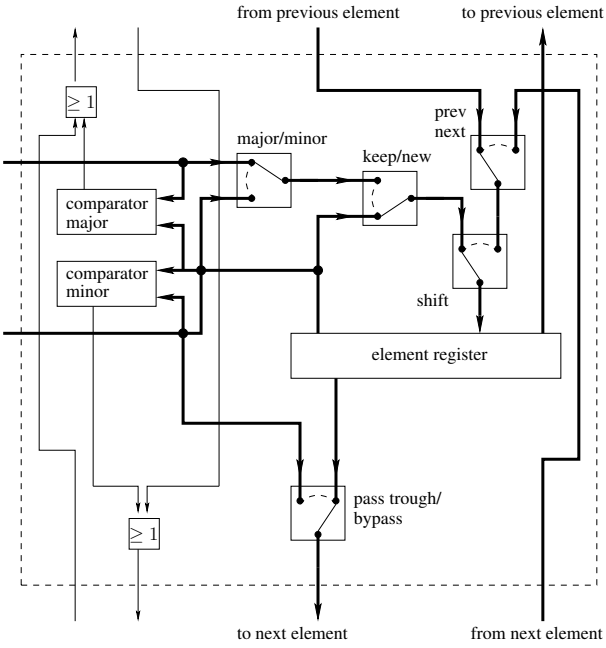


Fig. 5. Priority queue element

The algorithm terminates if the maximum possible path length is reached. The stored path in the top element is the decoded codeword. In the practical implementation an iteration counter will terminate after a determined maximum number of iterations. This abort can be used to mark this decoded GC column as a erasure symbol for the outer RS code $\mathcal{A}$.

C. Supercode Extension

In order to decode supercodes, the following extensions have to be implemented. First for each supercode $\mathcal{D}_i$ a distinct ROM is needed which represents its trellis. The metric calculation has to take all trellis branches of each $\mathcal{D}_i$ into account. Furthermore all node pointers have to be stored in the priority queue elements.

VI. Measurements and Comparison

In this section we present an FPGA implementation and C++ simulation results of the proposed soft input decoder and compare it with the hard input decoder presented in [4]. We first describe the GC code with its parameters. Next we compare the error correction performance and the throughput. Finally we present synthesis results to compare the implementation complexity.

For the GC code we use six levels with inner nested-BCH codes over $GF(2^6)$ and outer codes using $GF(2^9)$. In the first level the inner code can correct a single error and therefore six redundancy bits are needed. Thus the number of rows is $n_b = 6 \cdot 9 + 6 = 60$.

The hard input decoder uses algebraic decoding. It consists of the syndrome calculation, the Berlekamp–Massey algorithm (BMA), and the Chien search module. The soft input decoder is implemented as proposed in Section IV without the described SRAM improvement. It has two limitations. First, the length of the priority queue is limited to 64 elements. Furthermore the accuracy of the metric calculation is limited to 16 bits and we use 3-bit quantization of the input symbols.

We first compare the error correction performance of the GC code in different decoding modes and a long BCH code. The first decoding mode of the GC code is the hard input using algebraic decoding. We present two soft input modes where we use a 3-bit quantization and a floating point implementation that shows the maximum achievable gain in comparison with the hard input decoder. Note that the BCH code and GC code have different length. The BCH code was designed for blocks of 1k byte information, whereas the GC code is designed for 2k byte information blocks.

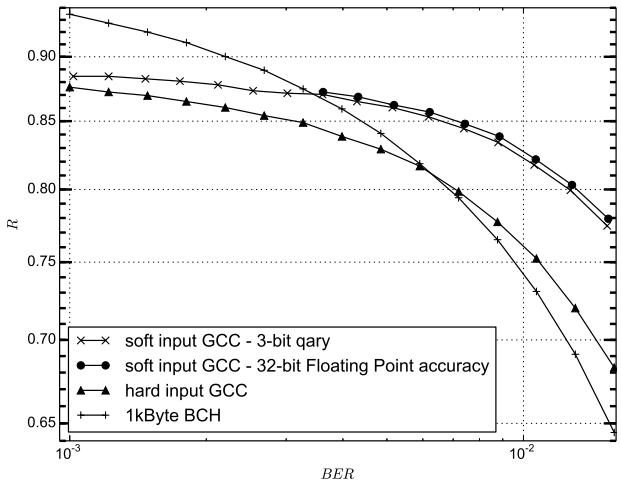


Fig. 6. Code rate versus bit error rate

Fig. 6 depicts the code rate versus bit error rate. The code rate was chosen in order to achieve the overall block error rate of 10^{-16}. As can be seen the hard input curve for the GC code is flatter than the curve for the BCH code. The GC decoder with 3-bit quantized soft information shows a significant gain.

Module	Slices	LUT
syndrome	118	234
iBMA	1 800	3 348
Chien search	1 266	2 304
Total	**3 184**	**5 886**
Stack alg.	**18 487**	**36 829**

Tab. I
Column Code FPGA Synthesis Size

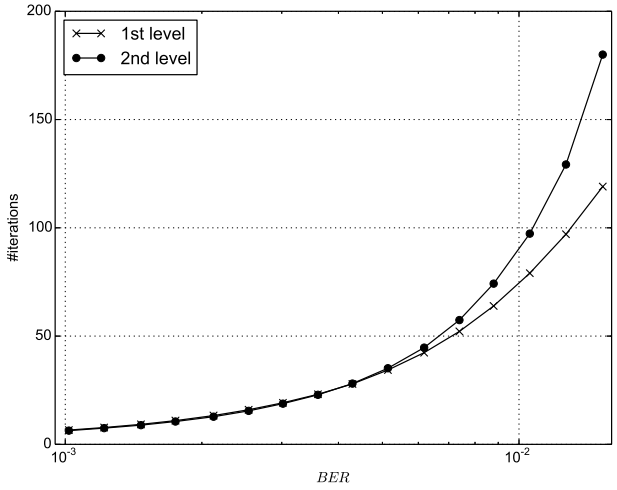


Fig. 7. Average number of iterations for the first and second level

The stack algorithm has a variable execution time depending on the error pattern. This algorithm needs at least 61 cycles to traverse the entire trellis if no error occurred. This case can be omitted by checking whether the syndrome of a column word is zero. If no error is detected the soft decoding can be avoided and thus only a single cycle is needed. Fig. 7 compares the average number of cycles needed for the stack algorithm. It shows the dependency between the channel bit error rate and the computational complexity, i.e., fewer errors lead to fewer decoding cycles. Note that the algebraic hard input decoder needs four cycles for the first and six cycles for the second level.

Next we present FPGA synthesis result for the stack algorithm. The synthesis was performed with Xilinx ISE 14.4 and a Virtex-4 target device. Tab. VI shows the number of slices and LUT of the hard input and the soft input decoder with 3-bit quantization.

VII. Conclusion

In this work we have presented a soft input decoder for generalized concatenated codes. We have proposed a sequential decoding algorithm based on supercode trellises in order to

reduce the complexity of the soft input decoding. This implementation improves the error correction capability significantly compared with hard input decoding. The implementation of the stack algorithm is nine times large than the algebraic decoder. This complexity can be reduces by moving the majority of the register entries into an SRAM. Nevertheless, the proposed soft input decoding increases the overall complexity of the GC decoder only by 72% compared with the decoder presented in [12]. Consequently, the proposed method is a promising approach for soft input decoding of GC codes.

ACKNOWLEDGMENT

We thank Hyperstone GmbH, Konstanz for supporting this project. The German Federal Ministry of Research and Education (BMBF) supported the research for this article (03FH025IX5).

REFERENCES

[1] A. Fahrner, H. Griesser, R. Klarer, and V. Zyablov, "Low-complexity GEL codes for digital magnetic storage systems," *IEEE Transactions on Magnetics*, vol. 40, no. 4, pp. 3093–3095, July 2004.

[2] J. Freudenberger, U. Kaiser, and J. Spinner, "Concatenated code constructions for error correction in non-volatile memories," in *Int. Symposium on Signals, Systems, and Electronics (ISSSE), Potsdam*, Oct 2012, pp. 1–6.

[3] J. Freudenberger, J. Spinner, and S. Shavgulidze, "Generalized concatenated codes for correcting two-dimensional clusters of errors and independent errors," in *Int. Conference on Communication and Signal Processing (CSP), Castelldefels-Barcelona*, Feb. 2014, pp. 1–5.

[4] J. Spinner and J. Freudenberger, "Design and implementation of a pipelined decoder for generalized concatenated codes," in *Proceedings of 27th Symposium on Integrated Circuits and Systems Design (SBCCI), Aracaju, Brazil*, Sept 2014, pp. 1–16.

[5] I. Dumer, *Concatenated codes and their multilevel generalizations*. in Handbook of Coding Theory, Vol. II, Elsevier, Amsterdam, 1998.

[6] M. Bossert, *Channel coding for telecommunications*. Wiley, 1999.

[7] A. Neubauer, J. Freudenberger, and V. Kühn, *Coding Theory: Algorithms, Architectures and Applications*. John Wiley & Sons, 2007.

[8] L. Aguado and P. Farrell, "On hybrid stack decoding algorithms for block codes," *Information Theory, IEEE Transactions on*, vol. 44, no. 1, pp. 398–409, Jan 1998.

[9] J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Transactions on Information Theory*, vol. 24, no. 1, pp. 76–80, Jan 1978.

[10] J. Freudenberger and M. Bossert, "Maximum-likelihood decoding based on supercodes," in *Proc. 4th. International ITG Conference Source and Channel Coding*, Erlangen, Germany, Jan. 2004, pp. 185–190.

[11] J. Spinner, J. Freudenberger, and S. Shavgulidze, "Sequential decoding of binary block codes based on supercode trellises," in *1st BW-CAR Symposium on Information and Communication Systems (SInCom)*, Nov 2014, pp. 25–28.

[12] J. Spinner and J. Freudenberger, "Decoder architecture for generalized concatenated codes," *IET Circuits, Devices & Systems*, vol. 9, no. 5, pp. 328–335, 2015.

Filtering Probabilistic Depth Maps Received from a Focused Plenoptic Camera

Niclas Zeller, Franz Quint
Faculty of Electrical Engineering and Information Technology
Karlsruhe University of Applied Sciences
76133 Karlsruhe, Germany
Email: niclas.zeller@hs-karlsruhe.de,
franz.quint@hs-karlsruhe.de

Uwe Stilla
Department of Photogrammetry and Remote Sensing
Technische Universität München
80290 Munich, Germany
Email: stilla@tum.de

***Abstract*—This paper presents a filtering approach for semi-dense probabilistic depth map received from a focused plenoptic camera. In the probabilistic depth map each valid depth pixel contains, beside the depth value itself, a variance which gives a measure for the certainty of the estimated depth. This variance is used in a weighted filtering approach. Here, beside removing outliers and filling holes in the semi-dense depth map, pixel neighborhoods are modeled in a Markov-Random-Field (MRF). The presented approach is working in two steps, firstly a rough regularization is performed in each micro image separately and secondly, after projection to the virtual image space, another more precise regularization is performed. The presented filtering approach considers properties of the plenoptic imaging, like varying spatial resolution over object distance. Besides, it preserves discontinuities in the depth map. The algorithm aims for low complexity and due to its nature it can easily be implemented in parallel.**

I. Introduction

While a monocular camera captures light intensities only on a 2D sensor, a plenoptic camera gathers intensities in a sampled 4D light-field representation.

Even though this concept to gather 4D light-field information was invented already more than a century ago [1], [2], it took until the last decade to put the first plenoptic cameras on the market. One reason therefor is the high computational cost which has to be spend to process the recorded light-field information. Nevertheless, today there are several algorithms available to process the recorded 4D light-field information in, or at least close to, real-time.

Processing tasks last from subsequent refocusing, over super resolution imaging, up to depth estimation. Especially plenoptic camera based depth estimation and the fact, that a plenoptic camera records much more information about scene structures than a monocular camera arouse attention in the computer vision community. Here, possible applications are for instance Visual Odometry (VO) or Simultaneous Localization and Mapping (SLAM) but also gesture and pattern recognition.

This paper continues the work of our prior publication [3] where a probabilistic depth estimation approach for a focused plenoptic camera was developed. The method in [3] establishes a semi-dense depth map, which means that depth is only estimated for regions of high contrast. Here we present a post processing step to filter out outliers and stochastic noise to improve the quality of the depth map.

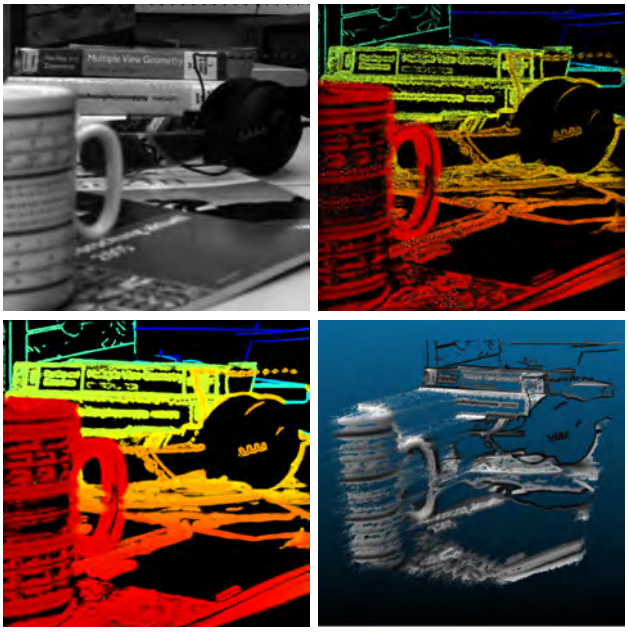


Fig. 1. Sample scene recorded by a focused plenoptic camera. Top left: Synthesized totally focused intensity image. Top right: Color coded unfiltered depth map. Bottom left: Color coded filtered depth map. Bottom right: Point cloud representation of the filtered depth map.

Due to the fact that we want to use the resulting depth information in Visual Odometry which is supposed to operate close to real-time, we were focusing on methods which have low complexity and can be implemented in an efficient manner.

In this article, we first briefly describe the concept of a focused plenoptic camera (Section II) which is requisite to understand our depth estimation approach. This approach is presented succinctly in Section III. Our filtering method based on the probabilistic depth map is introduced in Section IV. Section V shows how based on the estimated depth an intensity image with a very high depth of field (DOF) (totally focused image) can be synthesized. Section VI presents the evaluation of the filtering approach and Section VII concludes our work.

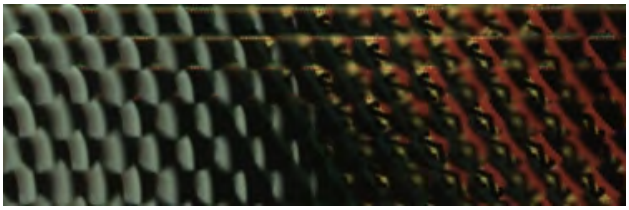


Fig. 2. Subsection of the raw image recorded by a focused plenoptic camera. The raw image consists of thousands of circular micro image arranged on a hexagonal grid.

II. Concept of the Focused Plenoptic Camera

A focused plenoptic camera (plenoptic camera 2.0) [4], [5], which the presented method relies on, slightly differs from a traditional, unfocused plenoptic camera [6], [7]. Nevertheless, both concepts are based on a micro lens array (MLA) which is placed in front of the sensor.

The main advantage of a focused plenoptic camera is that it produces a focused micro image of a subsection of the scene under each micro lens, as can be seen in the recorded sensor image in Figure 2. Thereby a higher spatial image resolution is achieve than with an unfocused plenoptic camera, where each micro lens just represents one spatial sample.

A focused plenoptic camera can be realized in two different setups [8], [4], the Keplerian and the Galilean mode. In the Keplerian mode the light-field sensor (image sensor and MLA) is placed in a distance further than the image distance b_L to the main lens of the camera, while in the Galilean mode the light-field sensor is placed closer than b_L to the main lens.

Figure 3 shows the interior of a focused plenoptic camera based on the Galilean setup. Here D_L defines the aperture of the main lens and D_M the one of a micro lens. b_L represents the image distance of the projected main lens image (virtual image). The relation between the image distance b_L and the object distance a_L is defined, dependent on the focal length f_L, by the thin lens equation:

$$\frac{1}{f_L} = \frac{1}{a_L} + \frac{1}{b_L} \tag{1}$$

In the following we will discuss only the Galilean mode. Nevertheless, similar assumptions can be made for the Keplerian mode.

In our research we are working with a Raytrix camera [5], which is a focused plenoptic camera operating in the Galilean mode. Besides, a Raytrix camera has as distinct feature an MLA that consist of three different types of micro lenses which are arranged in a hexagonal grid. Each type of micro lens has a different focal length and thus focuses a different image distance b_L and respectively object distance a_L on the sensor. The three focal lengths are chosen such, that the corresponding DOFs are just adjacent. Thus, the DOF of the camera is extended. Due to overlap of the micro images it can be assured, that each virtual image point occurs focused in at least one micro image.

From Figure 3 one can see, that a certain object point is projected to multiple micro images on the sensor. Thus, by finding corresponding points in the micro images, the distance b between MLA and the respective virtual image point can be calculated by triangulation:

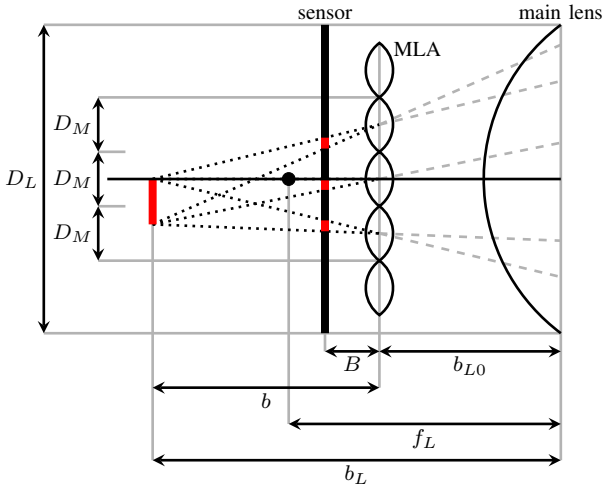


Fig. 3. Optical path inside a focused plenoptic camera based on the Galilean configuration. The MLA and the image sensor lie in front of the virtual image created by the main lens. A virtual image point in distance b behind the MLA results in multiple focused micro images on the sensor.

$$b = \frac{d \cdot B}{p_x} \tag{2}$$

Here, d represents the baseline distance between two micro lenses, B the distance between MLA and sensor and p_x the measured disparity between the corresponding points in the micro images. Since points with a large distance b are seen from micro lenses which are further apart, the lager baseline distance d can be used to improve the depth estimate. A detailed derivation of this equation can be found in [9]. Prior to a metric calibration, the distance B is not known precisely and thus the depth estimate is defined relative to this distance:

$$v = \frac{b}{B} = \frac{d}{p_x} \tag{3}$$

The distance v is called the virtual depth subsequently [5]. Besides, we defined the inverse virtual depth $z = v^{-1}$, which will be used in the following sections:

$$z = \frac{1}{v} = \frac{p_x}{d} \tag{4}$$

III. Probabilistic Depth Estimation

This section presents our probabilistic virtual depth estimation algorithm, originally published in [3]. The prior section showed that depth estimation in a focused plenoptic camera can be solved by finding pixel correspondences between micro images. Due to the high overlap of the micro images, this task can be considered as multi-view stereo problem. Since the focus of our approach lies on real-time applicability, the presented algorithm searches for corresponding pixel pairs based on local criteria. To combine all these correspondences

in one depth map, for each depth observation an uncertainty measure is defined which is used to establish a probabilistic depth map, as introduced in [10].

A. Probabilistic Virtual Depth

From eq. (4) one can see, that the inverse virtual depth z is proportional to the estimated disparity p_x. The disparity p_x of a point in two micro images is estimated based on intensity error minimization along the epipolar line. Thus, the sensor noise is considered to be the main error source effecting this estimate. Since the sensor noise is usually modeled as additive white Gaussian noise (AWGN), we also consider the estimated disparity p_x and hence also the inverse virtual depth to be a Gaussian distributed random variable Z, defined by the mean z and the variance σ_z^2.

$$f_Z(x) = \frac{1}{\sqrt{2\pi}\sigma_z} e^{-\frac{(x-z)^2}{2\sigma_z^2}} \tag{5}$$

In [3] it is derived in detail how the inverse virtual depth variance σ_z^2 can be obtained.

B. Virtual Depth Observation

For depth estimation we want to establish stereo correspondences between any possible pairs of micro lenses. Thus, a graph of baselines is build which defines the micro image pairs used for stereo matching. For a certain micro lens this graph contains the baselines to all micro lenses right to the micro lens of interest. In that way it is assured that each pixel correspondence is established only once. Figure 4 shows five sample baselines, one for each of the five shortest baseline distances.

Each baseline in the graph is defined by its length d as well as its 2D orientation on the MLA plane $\boldsymbol{e}_p = (e_{px}, e_{py})^T$. Since the micro images are considered to be rectified, the orientation vector of the baseline is equivalent to that of the epipolar line of any pixel under the micro lens of interest.

The inverse virtual depth estimation is performed for each pixel in the raw image $p_R := \boldsymbol{x}_R = (x_R, y_R)^T$ which lies under a micro lens and has a certain intensity gradient along the epipolar line $\boldsymbol{e}_p$. For each pixels multiple stereo observations are estimated starting from the shortest baseline up to the longest possible baseline. This is done, since for a short baseline it is more likely to find unambiguous correspondences, while longer baselines improve the precision. Thus, for each observation the prior estimates are used to limit the search range.

C. Merging Inverse Virtual Depth Observations

The prior Section described already that for a pixel p_R in the raw image multiple inverse virtual depth observations can be obtained. Each new observation is incorporated into a inverse virtual depth hypothesis similar to the update step in a Kalman filter. Here, the new inverse virtual depth hypothesis $\mathcal{N}(z, \sigma_z^2)$ results from the hypothesis prior the observation $\mathcal{N}(z_p, \sigma_p^2)$ and the new observation $\mathcal{N}(z_o, \sigma_o^2)$ as follows:

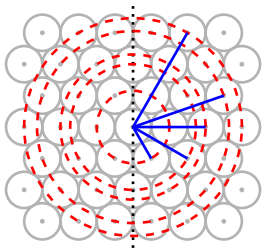


Fig. 4. Five shortest baseline distances in a hexagonal micro lens grid. For a micro lens stereo matching is only performed with neighbors for which the baseline angle ϕ is in the range $-90° \leq \phi < 90°$.

$$\mathcal{N}(z, \sigma_z^2) = \mathcal{N}\left(\frac{\sigma_p^2 \cdot z_o + \sigma_o^2 \cdot z_p}{\sigma_p^2 + \sigma_o^2}, \frac{\sigma_p^2 \cdot \sigma_o^2}{\sigma_p^2 + \sigma_o^2}\right) \tag{6}$$

IV. Post Processing of Probabilistic Depth Map

In the probabilistic depth estimation algorithm pixels correspondences are found only based on local criteria. Besides, each pixel is processed separately without considering a larger neighborhood. Thus, the estimated inverse virtual depth values z in the micro images can be considered to be more or less uncorrelated. This is actually not the case for depth maps of real scenes, where neighboring depth pixels usually are highly correlated. Thus, in this post processing step we model the connection between pixels in a certain neighborhood by a Markov-Random-Field (MRF).

The regularization is done in several steps. First outlier removal and hole filling is performed in each micro image separately (Sec. IV-A). Afterwards the pixels from the micro images are projected back into the virtual image space which is created by the main lens projection (Sec. IV-B). Here again outlier removal and hole filling is performed. Finally the depth map is updated based on an MRF structure, using all inverse virtual depth pixel hypothesizes within a certain neighborhood (Sec. IV-C).

A. Removing Outliers and Filling Holes in Micro Images

1) Removing Outliers: Due to ambiguous structures in the micro images it happens that wrong correspondences are established between pixels in the micro images. Thus, in a first step pixels which are outliers with respect to their neighborhood are removed. For each valid depth pixel in the raw image $p_R^{(i)}$, with the depth hypothesis $\mathcal{N}(z_i, \sigma_{zi}^2)$, an average inverse virtual depth $\overline{z_i}$ and a corresponding variance $\overline{\sigma_{zi}^2}$ of the valid depth pixels N_{Rvalid} within the neighborhood $N_R^{(i)}$ is defined:

$$\overline{z_i} = \frac{\sum_{k \in N_x^{(i)}, k \neq i} z_k \cdot \left(\sigma_{zk}^2\right)^{-1}}{\sum_{k \in N_x^{(i)}, k \neq i} \left(\sigma_{zk}^2\right)^{-1}} \tag{7}$$

$$\overline{\sigma_{zi}^2} = \frac{|N_x^{(i)}| - 1}{\sum_{k \in N_x^{(i)}, k \neq i} \left(\sigma_{zk}^2\right)^{-1}} \tag{8}$$

In eq. (8) $N_x^{(i)}$ defines the intersection between the set of neighborhood pixels $N_R^{(i)}$ of $p_R^{(i)}$ and the set of valid depth pixels N_{Rvalid} ($N_x^{(i)} = N_R^{(i)} \cap N_{Rvalid}$). Besides $|N_x^{(i)}|$ is the cardinality of the set $N_x^{(i)}$ and defines the number of elements in the set. Eq. (7) and (8) are actually quite similar to the definition of z and σ_z^2 in eq. (6). The only difference is, that the sum of the inverse variances is multiplied by the number of valid neighbors ($|N_x^{(i)}| - 1$).

Each pixel $p_R^{(i)}$ that has a inverse virtual depth estimate z_i which does not satisfy the following condition is classified as outlier:

$$(z_i - \overline{z_i})^2 \leq 4 \cdot \overline{\sigma_{zi}^2} \tag{9}$$

For the experiments in Section VI we defined a squared neighborhood of $5\,\text{pixel} \times 5\,\text{pixel}$.

2) Filling Holes: After removing the outlier in the micro images, pixels which have enough gradient but no valid depth estimate are filled based on the neighboring pixels. Therefore, again the average inverse virtual depth $\overline{z_i}$ within a neighborhood region is calculated based on eq. (7). The inverse virtual depth $\overline{z_i}$ gives the new depth value for the invalid pixel $p_R^{(i)}$, while the corresponding variance σ_{zi}^2 is initialized to some predefined high value. By setting the initial variance to some high value, these interpolated depth values can not negatively effect the later regularization.

B. Projecting Micro Images into Virtual Image Space

After removing outliers and filling holes in the micro images, all micro image pixels which have a valid depth hypothesis are projected into the virtual image space. We define the virtual image space similar to [3] by the pixel coordinates x_V and y_V and the virtual depth $v = z^{-1}$ ($\boldsymbol{x}_V = (x_V, y_V, v)^T$).

The virtual image coordinates x_V and y_V are calculated based on a central perspective projecting thru the corresponding micro lens as follows:

$$x_V = (x_R - c_x)v + c_x \tag{10}$$

$$y_V = (y_R - c_y)v + c_y \tag{11}$$

Here c_x and c_y is the center of the micro lens under which the pixel p_R with the coordinates $\boldsymbol{x}_R$ lies.

As described in Section II, a virtual image point is projected to multiple micro images. Vise verse, when performing the back projection to the virtual image space, multiple raw image pixels can fall on the same virtual image pixel and thus have to be merged. This is done similar to the merging step in Section III-C (eq. (6)).

C. Regularization of Virtual Image

In the virtual image space firstly outliers which were not detected in the micro images are removed. Besides, the inverse virtual depth values are smoothed, while considering the imaging concept of the camera.

For the virtual image space regularization again for each pixel $p_V^{(i)} := \boldsymbol{x}_V^{(i)}$ a neighborhood region $N_V^{(i)}$ is defined. Each micro lens performs a central perspective projection and thus, objects with a high virtual depth occur smaller on the sensor than objects with a small virtual depth. Besides, virtual image points with a high virtual depth are observed by more micro lenses. Thus, vise verse back projected virtual image regions with a high virtual depth consist of more points which are spread over a larger region. Hence, for the virtual image regularization, the size of the neighborhood is defined dependent on the virtual depth v_i of the pixel of interest $p_V^{(i)}$. For each pixel $p_V^{(i)}$ a radius $r(v_i)$ is defined as follows:

$$r(v) = \lceil n \cdot v_i \rceil \tag{12}$$

Here n defines some constant parameter. For lower complexity in calculation, the radius $r(v_i)$ defines the maximum allowed Chebyshev distance L_∞ to the pixel $p_V^{(i)}$ instead of the Euclidean distance. A pixel $p_V^{(k)}$ for instance has the Chebyshev distance $L_\infty^{(k)}$ to the pixel $p_V^{(i)}$ defined as follows:

$$L_\infty^{(k)} = \max\left(|x_V^{(i)} - x_V^{(k)}|, |y_V^{(i)} - y_V^{(k)}|\right) \tag{13}$$

Thus, a squared neighborhood $N_V^{(i)}$ around $p_V^{(i)}$ is defined.

1) Removing Outliers: For removing outliers in the virtual image, similar to the micro images, the mean inverse virtual depth $\overline{z_i}$ (eq. (7)) and the mean inverse virtual depth variance $\overline{\sigma_{zi}^2}$ (eq. (8)) of valid depth pixels within the neighborhood region $N_V^{(i)}$ is defined:

$$N_x^{(i)} = N_V^{(i)} \cap N_{Vvalid} \tag{14}$$

Here, N_{Vvalid} is the set of all pixels p_V which have a valid depth estimate. Nevertheless, besides fulfilling the condition defined in eq. (9) a pixel $p_V^{(i)}$ has to have a density of valid depth pixels in its neighborhood above a certain threshold T_D .

$$T_D \leq \frac{|N_x^{(i)}|}{|N_V^{(i)}|} \tag{15}$$

In the following experiments the minimum density was set to $T_D = 0.25$.

2) Filling Holes: At that point there are no intensities available for the virtual image. Thus, pixel validity can not be defined based on the pixel's intensity gradient, as it is done in the raw image. Instead, each pixel in the virtual image p_V which has at least one direct neighbor with a valid depth hypothesis is filled by a weighted average of neighboring pixels, defined similar to eq. (7).

3) MRF based Noise Reduction: In a final step the actual correspondence between neighboring pixels in the virtual image is established by modeling the neighborhoods in an MRF. Here the same neighborhood $N_V^{(i)}$ as for outlier removal is defined.

In the MRF for a neighborhood $N_V^{(i)}$ two different states are defined to handle discontinuities in the depth map. Thus, either a pixel $p_V^{(k)}$ in the neighborhood $N_V^{(i)}$ belongs to the same object as $p_V^{(i)}$ or to a different object. These two objects can be considered as foreground and background.

Besides, a weighting function $w(d)$ is defined which models the lower correlation between further apart pixels. Here d defines the euclidean distance between two pixels. In our approach we use a Gaussian curve as weighting function, where the standard deviation σ_w is proportional to v.

$$w(d) = e^{-\frac{d^2}{2\sigma_w^2}} \tag{16}$$

To model the two different states (foreground and background), the neighborhood is divided in two different sets of valid depth pixels, N_{sim} and N_{diff}. Here N_{sim} are all valid pixels which have a depth value similar to $p_V^{(i)}$ (including $p_V^{(i)}$) and N_{diff} contains all valid pixels with a depth value different to $p_V^{(i)}$. The updated depth value of the pixel $p_V^{(i)}$ is then calculated based on the larger one of the two sets as follows:

$$z_i = \frac{\sum_{k \in N_x} z_k \cdot \frac{w_k}{\sigma_{zk}^2}}{\sum_{k \in N_x} \frac{w_k}{\sigma_{zk}^2}} \tag{17}$$

$$\sigma_{zi}^2 = \frac{\sum_{k \in N_x} w_k}{\sum_{k \in N_x} \frac{w_k}{\sigma_{zk}^2}} \tag{18}$$

Here, N_x is the one of N_{sim} and N_{diff} with the higher cardinality.

$$N_x = \begin{cases} N_{sim} & \text{if } |N_{diff}| < |N_{sim}|, \\ N_{diff} & \text{else.} \end{cases} \tag{19}$$

w_k is defined as follows:

$$w_k = w(d_k) = w\left(\|(x_V^{(i)}, y_V^{(i)})^T - (x_V^{(k)}, y_V^{(k)})^T\|\right) \tag{20}$$

V. Intensity Image Synthesis

Additionally to the virtual depth map an intensity image is synthesized. This is done as described in [5]. For each point $p_V^{(i)}$, based on its inverse virtual depth estimate z_i a radius R_i is defined.

$$R_i = \frac{D_M}{2 \cdot z_i} \tag{21}$$

Here, D_M is the diameter of a micro lens as shown in Figure 3. The radius R_i defines the region around the pixel $p_V^{(i)}$ in which micro lenses still see this virtual image point. Hence, the point $p_V^{(i)}$ is back projected to all micro images for which the center lies within this radius around the pixel. Here only the micro lens types are considered in which the virtual image point is in focus. Based on the corresponding interpolated intensity values a weighted mean is calculated.

$$I(p_i) = \frac{\sum_k I_k \cdot h_k}{\sum_k h_k} \tag{22}$$

Here h_k are the respective intensity values received from a recorded white image. By calculating a weighted mean using the information of the white image for each pixel the optimum signal-to-noise ratio (SNR) is obtained.

For pixels p_V in the virtual image which have no depth estimate it is satisfactory to assume an approximate depth value to synthesize its intensity. This can be done since such pixels anyways lie in a neighborhood of little texture and thus all pixels in the neighborhood have similar intensities.

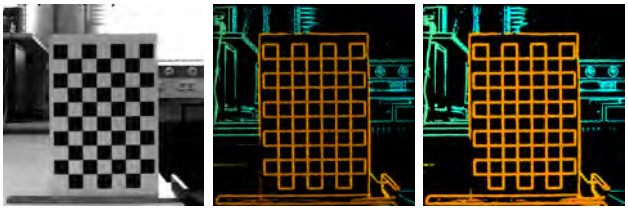


Fig. 5. Planar chessboard target used for quantitative evaluation. Left: Totally focused intensity image. Center: Color coded depth map before filtering. Right: Color coded depth map after filtering.

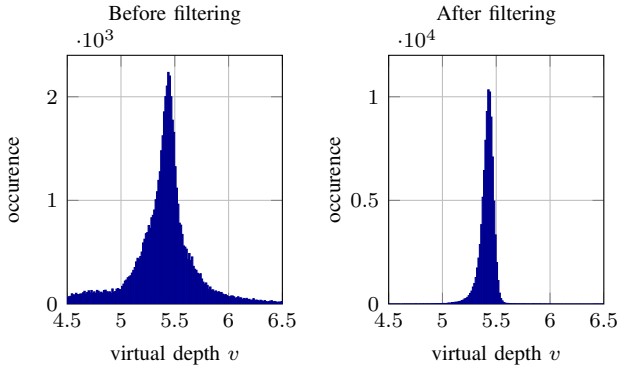


Fig. 6. Histogram of virtual depth v across chessboard target.

VI. Results

In this section we evaluate the presented filtering method based on artificial as well as real scenes. In a first evaluation we try to measure the filtering results in numbers, as presented in Section VI-A, while Section VI-B presents a qualitative evaluation.

A. Quantitative Evaluation

For a quantitative evaluation we used a planar target as ground truth which was placed in front of the plenoptic camera, as shown in Figure 5. Here the virtual depth values across the planar target were evaluated and compared before and after filtering. Figure 6 shows the histograms of the virtual depth v before and after filtering.

Besides, Table I shows some calculated statistics for the virtual depth v across the chessboard target (mean $\overline{v}$, median $\tilde{v}$, standard deviation σ_v).

As one can see, the standard deviation σ_v is highly reduce by the filtering. Furthermore, the arithmetic mean $\overline{v}$ is effected quite a lot by the filtering. This is because the virtual depth is

	mean $\overline{v}$	median $\tilde{v}$	std. dev. σ_v
Before filtering	5.667	5.430	1.304
After filtering	5.413	5.423	0.071

TABLE I
Statistics calculated for virtual depth across chessboard plane.

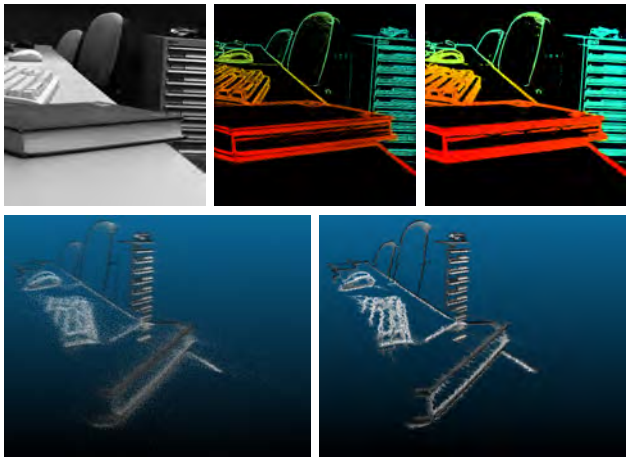


Fig. 7. Real scene recorded by a focused plenoptic camera. Top left: Synthesized totally focused intensity image. Top center: Color coded unfiltered depth map. Top right: Color coded filtered depth map. Bottom left: Point cloud representation of the unfiltered depth map. Bottom right: Point cloud representation of the filtered depth map.

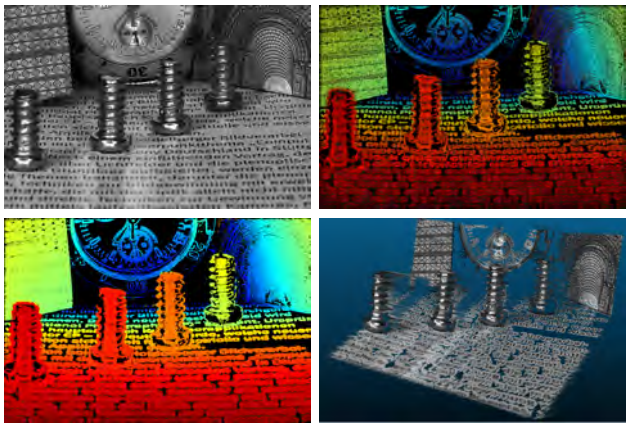


Fig. 8. Raytrix sample scene [11]. Top left: Synthesized totally focused intensity image. Top right: Color coded unfiltered depth map. Bottom left: Color coded filtered depth map. Bottom right: Point cloud representation of the filtered depth map.

not symmetrically distributed but the inverse virtual depth.This mapping from z to v has the effect, that the mean is shifted away from the distributions maximum towards higher virtual depth values. Nevertheless, the median $\tilde{v}$ stays more or less constant.

B. Qualitative Evaluation

In this part of the evaluation some real scenes are presented for which the depth was estimated and filtering was applied. Figure 1 and Figure 7 show two real scenes which were recorded with a Raytrix R5 camera. In Figure 8 a sample scene from Raytrix [11] is shown.

The color coded depth maps show quite well how holes in the depth map are filled and how pixel in a neighborhood of low density are removed. From Figure 7 one can see that this sometimes also removes some valid structures form far away regions (e.g. at the tool cabinet in the back). This could be avoided by adjusting the parameters. Nevertheless, this also might cause some unwanted outliers to be kept.

The bottom row in Figure 7 shows a comparison of the point clouds before and after filtering. These point clouds nicely indicate how the noise is reduced but discontinuities are kept.

VII. Conclusion

In this paper we presented a method to filter semi-dense probabilistic virtual depth maps which were estimated based on a focused plenoptic camera. Besides removing outliers and filling holes, the approach uses an MRF to smooth the depth map, while preserving discontinuities.

The presented results show high improvement in comparison to the unfiltered depth information. Noise is highly reduced while most of the details are kept.

The presented method was developed in such manner, that it later can be implemented as real-time post processing for instance on a graphic processor unit (GPU). The properties of the resulting depth map are such that it supplies reliable depth information which later can be used in a plenoptic camera based Visual Odometry.

In future one could think of including intensity information for filtering the virtual image space, similar to the micro images. Nevertheless, since the correct intensity of a pixel relies on a correct virtual depth estimate, the problem becomes more complex.

Acknowledgment

This research is funded by the Federal Ministry of Education and Research of Germany in its program ”IKT 2020 Research for Innovation”.

References

[1] F. E. Ives, “Parallax stereogram and process of making same,” USA Patent US725 567, 04 14, 1903.

[2] G. Lippmann, “Epreuves reversibles. photographies integrales,” *Comptes Rendus De l’Academie Des Sciences De Paris*, vol. 146, pp. 446–451, 1908.

[3] N. Zeller, F. Quint, and U. Stilla, “Establishing a probabilistic depth map from focused plenoptic cameras,” in *Proc. International Conference on 3D Vision (3DV)*, 2015, pp. 91–99.

[4] A. Lumsdaine and T. Georgiev, “The focused plenoptic camera,” in *Proc. IEEE International Conference on Computational Photography (ICCP)*, San Francisco, CA, April 2009, pp. 1–8.

[5] C. Perwaß and L. Wietzke, “Single lens 3d-camera with extended depth-of-field,” in *Proc. SPIE 8291, Human Vision and Electronic Imaging XVII*, Burlingame, California, USA, January 2012.

[6] E. H. Adelson and J. Y. A. Wang, “Single lens stereo with a plenoptic camera,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 99–106, February 1992.

[7] R. Ng, M. Levoy, M. Brédif, G. Guval, M. Horowitz, and P. Hanrahan, “Light field photography with a hand-held plenoptic camera,” Stanford University, Computer Sciences, CSTR, Tech. Rep., 05 2005.

[8] A. Lumsdaine and T. Georgiev, “Full resolution lightfield rendering,” Adobe Systems, Inc., Tech. Rep., 2008.

[9] N. Zeller, F. Quint, and U. Stilla, “Calibration and accuracy analysis of a focused plenoptic camera,” *ISPRS Annals of Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. II-3, pp. 205–212, 09 2014.

[10] J. Engel, J. Sturm, and D. Cremers, “Semi-dense visual odometry for a monocular camera,” in *Proc. IEEE International Conference on Computer Vision (ICCV)*, Dec 2013, pp. 1449–1456.

[11] Raytrix GmbH, last accessed: July 30, 2015. [Online]. Available: http://www.raytrix.de/

Building Distributed and Intelligent Systems by the Dynamic Embedment of Peer-specific Resource Capabilities and Rich Environment Models

Maximilian Engelsberger
Institute of Smart Systems and Services
Pforzheim University
Pforzheim, Germany +49 7231–286851
Email: Maximilian.Engelsberger@hs-pforzheim.de

Thomas Greiner
Institute of Smart Systems and Services
Pforzheim University
Pforzheim, Germany +49 7231–286689
Email: Thomas.Greiner@hs-pforzheim.de

***Abstract*—In this paper a new architectural method for the dynamic embedment of resources and models into system nodes at run-time is proposed. One approach for realizing this is the dynamic embedment of peer-specific resource capabilites, such as I/O-resources or cloud-based resources, as well as rich cyber and physical environment models between networked system nodes by an adapted multi-tier architecture. The method considers the latest QoS-metrics of the communication link and ensures a reliable basic system operation, even when the network fails completely. Generic selection-algorithms for a metrics-based evaluation of the application-specific sub-algorithms and model-components are described. An example implementation and performance evaluation of the proposed architecture is given with an application from the field out of the renewable energies. Therefore, a quantitative evaluation of the round-trip-time (RTT) of the control-loop, as well as trend results of the costs are given by an economies-of-scale-model.**

I. Introduction

One of the main characteristics of a cyber-physical system (CPS) is the complexity of the system resulting from its networked components: It consists of two or more system nodes, each equipped with its own sensors and actuators, interacting with the physical environment [1]–[3]. They are interconnected over local and/or wide area networks - like the Internet - with other nodes, processes or humans (see Figure 1) [4], [5]. Cloud-based resources can be integral part of the system as well, in order to provide practically endless computing capacities and storage space to the other system nodes [6]–[10]. A main distinguishing feature to already established types of embedded systems is the openness of the communication network, to other applications [11]. That means there can be various participating subsystems from different vendors on the same communication infrastructure and they may implement applications from two or more different application domains [11]. Such an environment, which is not under control of one single person or institution, will never be guaranteed static [6], [11]. A continuous complementation of sub-algorithms and model-components becomes essential to upgrade functionality and react to new system requirements and dynamic changing environment conditions. This blurs the old-days strictly defined transition between design-time and run-time (see Figure 2). Situation-related resource sharing, enabled through networked system components, becomes a key technique for diverse development efforts of future cyber-physical systems (CPS). The following chapters discuss a method of a dynamic embedment of resource capacities and rich environment models (REMs) within a CPS, in order to enable such complementation processes in a comfortable and economic way through a high-level servicing point.

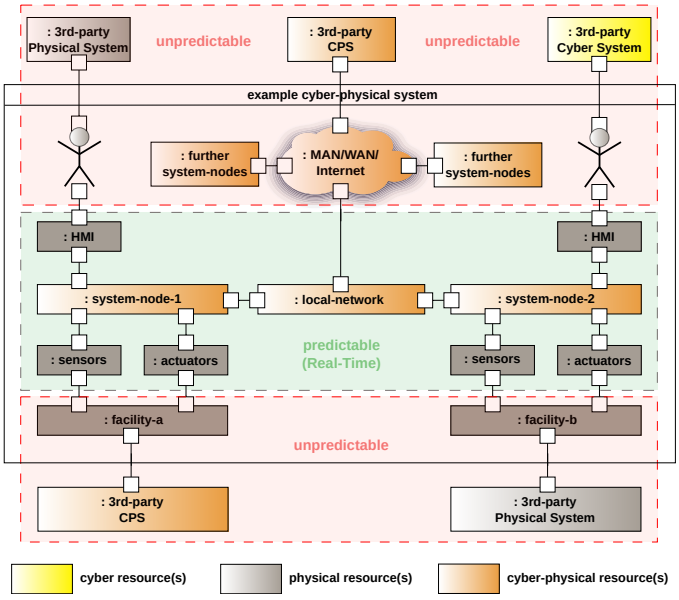


Fig. 1. Generic model of a cyber-physical system with scope of a resource-based view [4], [5]

II. Related Work

In [10] a new architectural approach of cloud-based cyber-physical control systems was discussed. The proposed software architecture came with an adapted multi-tier- and multi-layer-architecture [12], which helps to master the heterogeneities in between hardware architectures, operating system platforms and programming languages/APIs (see Figure 3). A solution for the seamless integration of sensors and actuators into cloud-based algorithms, as well as the mitigation of the vendor lock-in effect between different cloud-service providers and system vendors, was shown. For this reason, a method for a flexible

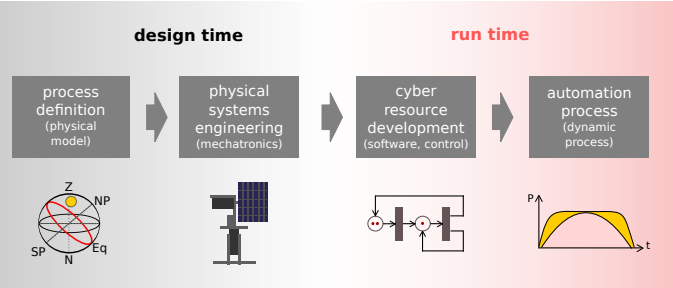


Fig. 2. Dynamic CPS workflow showing the blur of design-time and run-time by the application example introduced in chapter V

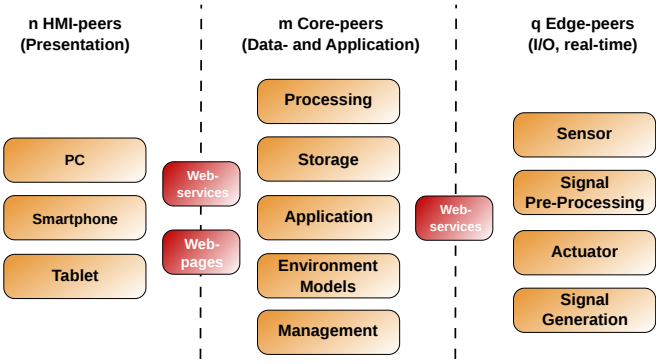


Fig. 3. The multi-tier architecture including HMI-peers, Core-peers and Edge-peers

application of the Software-as-a-Service (SaaS) [13] and on-premises model, was explained.

III. State of the Art

The adapted multi-tier architecture from [10], as seen in Figure 3, consists of three tier-types, which are redefined here as follows: A HMI-peer is a kind of peer like a PC, smartphone or tablet computer. It consumes and/or provides resources from the other peers. The Core-peers consists of cloud-based resources (Software-as-a-Service, SaaS) or on-premises resources. They are peers with huge processing- and storage capacities. That is why they are running complex algorithms or holding the rich environment models of the application. The Edge-peers come along with a minimal set of computing resources. They are mostly used as highly specialized Edge-peers to interface the physical environment with their I/O-capabilities (see Figure 1), and to provide and/or consume services from the overall CPS. In the adapted multi-tier-architecture the separate tiers are named "peers". That is because each of them can provide as well as consume resource services. Such a system architecture, where each participating node/tier can implement client- as well as server-functionalities, is called a Peer-to-Peer-Architecture (P2P) [14]. A P2P-Architecture is a key component in dynamically changing environments of cyber-physical systems, where resource capabilities and rich environment models shall be embedded over node-borders bilaterally. Resources are shared between tiers as Web-services using the REST-protocol (Representational State Transfer), no specific P2P-Protocol is needed at this time, because the systems' peers are considered using static addresses and every peer knows each other peers addresses.

IV. A New Architectural Method for the Dynamic Embedment of Peer-specific Resource Capabilities and Rich Environment Models

The following sections discuss the space for improvement of the adapted multi-tier architecture regarding the embedment of sub-algorithms and models or model-components dynamically at run-time by the evaluation of application-specific execution-metrics. This realizes two main advantages: The first is a dynamical increasing of the systems' fitness, depending of the metrics evaluation. That means if the metrics pass, more complex resources can be embedded during run-time and smarter control algorithms and models can be applied. This can lead to better control results. The second advantage is the creation of a high-level servicing point on the Core-Peers. That allows to implement complex algorithms and models in high-level programming languages like Java. Instead of changing fragile firmware code, it becomes possible to easily implement, test and maintain high-level code on the Core-Peers. The principal details of that architectural aspects are described below.

A. Dynamic Embedment of Resource Capabilities

The fundamental approach of the proposed method is to implement at least one very basic version of the application-specific control algorithm on the Edge-peers. That ensures a basic operation, independent from the network conditions. The more complex algorithms, which need more resource capacities for execution, are located on the Core-peers. The Core-peers can be realized as cloud-based resources. That makes them very scalable, if more processing capacitiy is needed. The Edge-peers are limited in their capacities and can be optimized for minimal costs or energy consumption. The idea is to embed the more complex algorithms in the Edge-peers processing flow dynamically over the network, if the preconditions are fulfilled. That allows more accurate results and a more adaptive behavior of the given automation task. Further reasons are the better utilization of available processing capacities and storage space on the Core-peers as well as the integration of more data-sources as they are available on the Edge-peers (see section IV-B). If the communication link to the Core-peers fails, the Edge-peers are able to perform their tasks in a a basic characteristic, without the need for a link to the Core-peers. For realizing this, the proposed method consists of a set of application-specific sub-algorithms which are ordered by their complexity in ascending order (see algorithm 1). Low complexity means basic characteristic of the algorithm but it still ensures a working system. High complexity means better control results but high resource demands respectively high execution dependancies. Each sub-algorithm is assigned to a metric, which evaluates its execution dependencies. The metrics are evaluated, started with the lowest complexity, if the metric pass, it jumps right to the next-higher complex sub-algorithm and checks its metric and so on. This may be QoS-metrics of the communication link between the Core- and Edge-peers for example, internal error- or fault-states, sensor thresholds etc. If a metric evaluation fails, the sub-algorithm with the next-lower complexity will be executed. An example implementation is given in section V-D and Figure 4.

Algorithm 1 Dynamic Embedment of Resource Capacities

Require: List of available sub-algorithms L_{sa}
while $control \neq aborted$ **do**
  actual sub-algorithm $A_{sa} := L_{sa}[0]$;
  for all sub-algorithm $S_a \in (L_{sa})$ **do**
    if assigned metric pass **then**
      $A_{sa} := S_a$;
    else
      keep A_{sa};
  Include A_{sa};
  Execute A_{sa};

Algorithm 2 Dynamic Embedment of Resources and Models

Require: List of available sub-algorithms L_{sa}
Require: List of available model-components L_{mc}
while $control \neq aborted$ **do**
  actual sub-algorithm $A_{sa} := L_{sa}[0]$;
  for all sub-algorithm $S_a \in L_{sa}$ **do**
    if assigned metric pass **then**
      $A_{sa} := S_a$;
      actual model-component $A_{mc} := L_{mc}[0]$;
      for all model-component $M_c \in L_{mc}$ **do**
        if assigned model metric pass **then**
          $A_{mc} := M_c$;
        else
          keep A_{mc};
    else
      keep A_{sa};
  Include A_{mc};
  Include A_{sa};
  Execute A_{sa};

B. Dynamic Embedment of Rich Environment Models

The proposed method of section IV-A can be extended to consider the availability of rich environment models (REMs) as well. A rich environment model may consist of cyber- as well as physical model components, describing the CPS' environment or parts of the system itself. Every control system needs some kind of environment model, in order to fulfill its application-specific control task. Such a model can be a very simple state machine or closed-loop control algorithm like a PID-control for example. Each model or model component is assigned to an application-specific evaluation metric as well. The Edge-peers of a CPS must implement at least one basic model by themselves, to keep the system operational, even in case of a network failure between the Edge- and Core-peers. But if the metrics are fine, the system can become fitter by the dynamic embedment of rich environment models from the Core-peers (see algorithm 2). Such a rich environment model can consist of several model-components, which are applied in the control system within the sub-algorithms. The more complex models or model-components are implemented on the Core-peers, because they are much bigger dimensioned than the Edge-peers. Third-party data-sources can be used to generate or update them continuously. Moreover new models or updates of models can be realized on a high-level servicing point on the Core-peers instead of implementing and rolling out complex firmware code on the Edge-peers and getting in

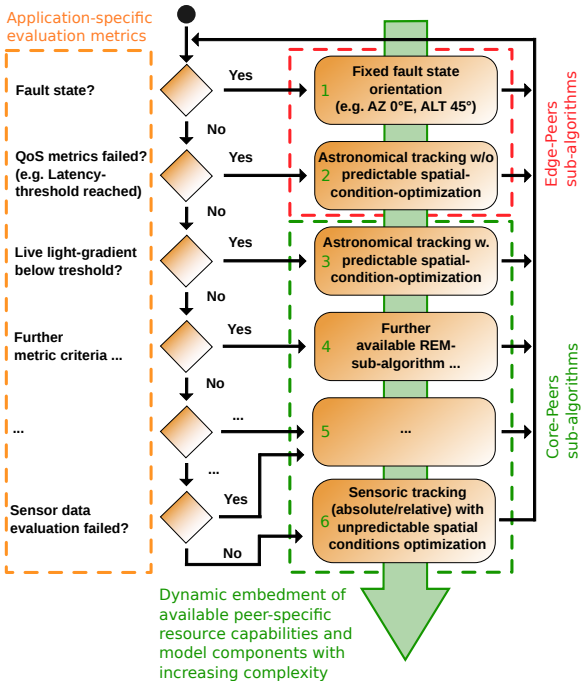


Fig. 4. Example of a dynamic smart tracking algorithm

struggle with the Edge-peers hardware limits. The dynamic embedment of rich environment models is a key feature to react on changing system components or on a changing system environment during run-time in an easy and economic way.

V. Application

The proposed method and architecture is practically implemented and validated with an application out of the field of renewable energies. The application addresses both advantages of the proposed method:

- Dynamic embedment of resources and REMs
- Realizing a high-level servicing point

A CPS is constructed, which shows the architectural principles through the realization of a smart photovoltaic facility [10]. The physical basic conditions and the method of resolution is described as follows.

A. Physical Basic Conditions

While earth is rotating, this leads to an apparent movement of the sun on the day's sky. The energy-output P_{out} of a photovoltaic module (PV module) depends on the surface which is illuminated by the sunlight. Therefore, a fix mounted PV module reaches its $P_{out,max}$ only once a day, for a very short period of time [10]. And this is only true, if the unpredictable spatial conditions like weather etc. are optimal as well.

B. Principles of Solar Tracking

A nearby solution is to move - track - the PV module with the sun, in order to keep the illuminated area at its maximum. The most simple approach is the astronomical

tracking, where the module's orientation is tracked by its geo-location and day/time [15]. One of the biggest disadvantages of this method is the disability of reacting to unpredictable environment conditions like cloudy sky areas (see Figure 6). The second basic approach is the sensoric tracking, where the current sky conditions can be considered [10], [15]. This method may result in very good results, if all conditions are fine and into very bad results, if they are not (see Figure 7). A high occurrence of short disturbances from the environment, like light reflections for example, can cause system instability and may lead to a negative energy balance ($P_{out} < 0$).

C. Pinciples of Smart Tracking

Both basic tracking methods could easily be implemented on a lightweight embedded control system, but they are not able to deal with a dynamically changing environment, like unpredictable weather situations, reflections and shadowing effects etc. Predictable and unpredictable disturbances are able to highly affect the systems' energy balance. A smarter approach is to combine the basic methods in an intelligent way, through a dynamic evaluation of sensor data and third-party data sources. This makes the system able to select the best method of a given geo-position, time/date and present environment conditions. Such evaluation metrics, models and algorithms are more complex and need probable higher computing capacities and more storage space. The ability of an incremental improvement and maintenance of such algorithms is desirable. That is why such an example algorithm is implemented using the proposed multi-tier architecture.

D. Example of a Smart Tracking Algorithm

The benefit of the proposed method for the solar tracking application is to have a basic tracking functionality on the Edge-Peers. With the availability of the network and the Core-Peers more sophisticated algorithms and models can be applied in order to make the tracking more situation-adaptive and efficient. The smart tracking algorithm gives an example implementation of the dynamic embedment of resource capabilities from Core-Nodes into Edge-Nodes as described in section IV-A and algorithm 2. In the following, an implementation of a smart tracking algorithm is described: A set of application-specific execution-metrics are defined and ordered ascending by their complexity. They are evaluated during each control cycle. The sub-algorithm with the highest complexity, which passes the evaluation, is executed (see line 4 in algorithm 1). As described in section IV-A, a dynamic reaction on changes inside the system or from the systems' environment gets possible. An example of a dynamic change inside the system may be a lack of QoS in between Edge-peers and Core-peers (see II). The exemplarily smart tracking algorithm is implemented in such a way, that the systems' basic operation keeps up and running, even with very bad QoS or a broken connection between Edge- and Core-peers. In fault state, the algorithm will point to a fixed fault state orientation, energy output will be as good as a fixed mounted PV module. If QoS is below a defined threshold (e.g. 150 ms of latency), a simple astronomical tracking without optimization of predictable spatial-conditions is executed (see Figure 4). If QoS is better, the external resources from the Core-peers are embedded into the Edge-peers functionality and an optimized astronomical tracking is executed (see Figure 4). In this example a first model component is applied, which considers date/time and geo-location respectively the time zone of the tracker. If the light-gradient (see next section) is reaching a certain threshold, the sensoric tracking, realizing an unpredictable spatial-condition optimization, is executed (see next section and Figure 8). The benefit is to have all this sub-algorithms, except the fault-state- and simple astronomical tracking-algorithm, located on the Core-peers. Here they are realized as a cloud-based resource running on top of a GNU/Linux operating system. The implementation of the Edge-peers firmeware is written in C99, the advanced and more complex sub-algorithms on the Core-peers are implemented in the high-level programming language Java.

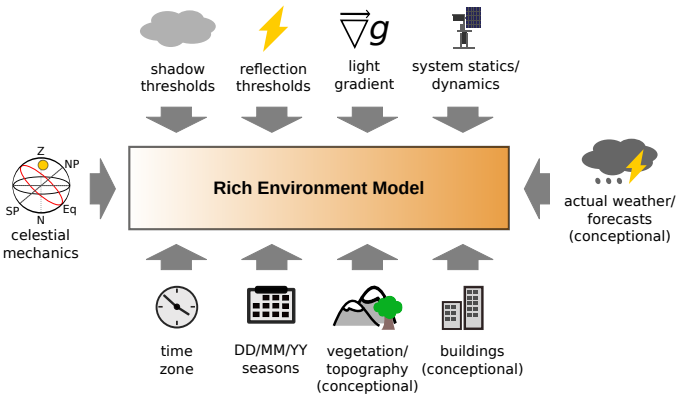


Fig. 5. Example of a rich environment model in the context of solar tracking which can continuously be updated and improved

E. Example of a Rich Environment Model

The rich environment model (REM) may consists of several model components, each describing a certain aspect of the system's environment or its interaction with it (see Figure 5). The model components are included in the smart tracking sub-algorithms and can complement the systems' ability to analyze and decide, dependent on their availability, which is evaluated by the respectively assigned metrics (see line 8 in algorithm 2). One part of the rich environment model of the proposed example application is the computation model of the live light-gradient, which will be described exemplarily below. It consists of the 2-dimensional calculation of the brightest light source in the sky, a set of thresholds, conversion and scaling formulas as well as the possible positions and orientations of the light sensors on the tracker. The models are implemented in C99 or Java, depending on their localization: On the Edge-peer or Core-peer. A basic overview about the live light-gradient is given below. The conversion and scaling algorithms are summerized as sensor signal preprocessing (see Figure 3). Everything which is not needed to be done on the Edge-peers, in order to fulfill the basic system operation, is implemented inside the REM on the core peer.

The live light-gradient describes the magnitude and direction of the brightest light-source in the sky, evaluated by the sensor data input, valid for each control cycle. The live light-gradient is calculated within a 2-dimensional scalar field, which represents the nonhomogeneous distribution of light in the sky. As an optimal result, the live light-gradient points to

the direction of the sun or - more precisely - to the energy source with the biggest magnitude. The scalar field is defined as the hemisphere represented by the sky. The 2-dimensional scalar field is sampled by the four light sensors, once in each control cycle. Four discrete initial values are generated as a representation of the continuous scalar field. This values are used for the component-by-component formation of the live light-gradient. For instance the difference between the azimuth-left and the azimuth-right signal and the altitude-up and altitude-down signal are calculated. These form the input values for the actual values of the semi-parallel working control algorithms. Hereby, the live light-gradient is calculated as the follows:

$$\overrightarrow{\nabla} g = \frac{\partial g}{\partial x_1} \cdot \hat{e}_1 + ... + \frac{\partial g}{\partial x_n} \cdot \hat{e}_n = \begin{pmatrix} \frac{\partial g}{\partial x_1} \\ ... \\ \frac{\partial g}{\partial x_n} \end{pmatrix} \quad (1)$$

The variables $\hat{e}_1$ and $\hat{e}_n$ are representing the unit vectors in the direction of the coordinate axis. In the implemented model there is no spherical transformation to the plane yet. This approach produces working results in good approximation. Therefore, for the present case in $\mathbb{R}^2$:

$$\overrightarrow{\nabla} g = \frac{\partial g}{\partial x} \cdot \hat{e}_x + \frac{\partial g}{\partial y} \cdot \hat{e}_y = \begin{pmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{pmatrix} \quad (2)$$

Where x is the azimuth- and y is the altitude measurement. A example with realistic sensor values could look like the follows: $\frac{\partial g}{\partial x} = 2400$ und $\frac{\partial g}{\partial y} = -1800$. From this follows:

$$\overrightarrow{\nabla} g = \frac{\partial g}{\partial x} \cdot \hat{e}_x + \frac{\partial g}{\partial y} \cdot \hat{e}_y = 2400 \cdot \hat{e}_x - 1800 \cdot \hat{e}_y = \begin{pmatrix} 2400 \\ -1800 \end{pmatrix} \quad (3)$$

Part of the live light-gradient model is a set of thresholds, which defines the activation levels of the control algorithms, in order to avoid frequent position changes of the tracker. Without such thresholds it can lead to an unstable behaviour of the gear controls. The thresholds are experimentally adjusted to +/- 10 mV at this time. The model and its evaluation metric are implemented on the Core-peer.

VI. Performance and Stability Evaluation

As a quantitative evaluation, a timing performance measurement is performed, which shows the full round-trip time (RTT) of the communication messages between edge- and core tiers [10]. The latency analysis is shown as a histogram, representing the percentages of transmissions, grouped by latency classes with a granularity of 5 ms (see Figure 9). The method works out well for the evaluated application. This is confirmed by the results from the quantitative timing measurements, where over 97 percent of all resource requests had a round-trip-time $\leq$ 45 ms. For more details about the used testcase of this series of measurements see [10]. The economies of scale model from [10] is improved. The model considers the basic costs c_b of the system as well as the scaling costs c_{sc} in a more detailed way, now. Each type of scaling costs take the different tier types from the adapted multi-tier architecture into account. Figure 10 shows the trend of costs with increasing number of nodes, changes and complexity per tier type. The data were generated by example cost-relation data.

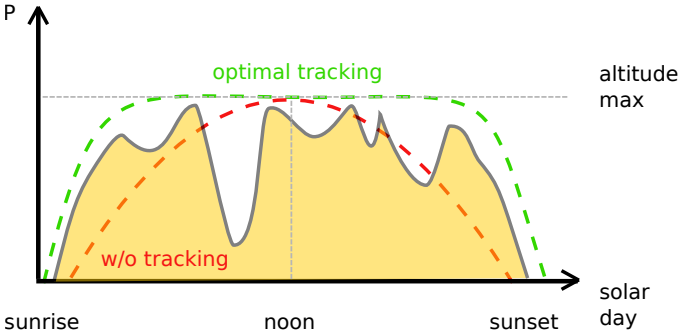


Fig. 6. Schematic power conversion with astronomical tracking without (un)predictable spatial conditions optimization and with (un)predictable spatial conditions (e.g. buildings, clouds)

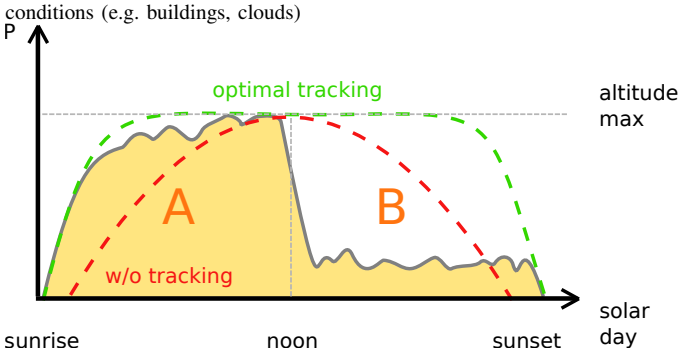


Fig. 7. Schematic power conversion with sensoric tracking (A: good conditions, B: High occurrence of short disturbances e.g. reflections etc.)

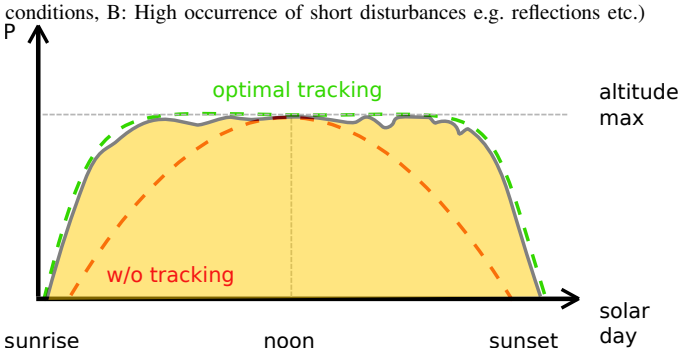


Fig. 8. Schematic power conversion with an adaptive smart tracking algorithm, considering predictable and unpredictable spatial conditions

$$c_t = c_b + c_{sc} \quad (4)$$

The parameters categories are the cost-numbers of system nodes p_{no}, the code changes P_{ch}, and the algorithm complexity p_{co}. The cost categories are named as follows: Total costs c_t, the basic costs c_b and the scaling costs c_{sc}. The scaling costs c_{sc} are composed by the sums of the weighted parameter-costs c_p for each of the different tier types.

$$c_{sc} = \sum_{x=0}^{n-1} p_x \cdot c_{p,x} + \sum_{y=0}^{m-1} p_y \cdot c_{p,y} + \sum_{z=0}^{q-1} p_z \cdot c_{p,z} \quad (5)$$

with $p_x \in P$ and $P = \{p_{ch}; p_{co}; p_{ti}\}$. The economic meaning of the method is confirmed by the economies of scale model which was printed with a set of example values (see

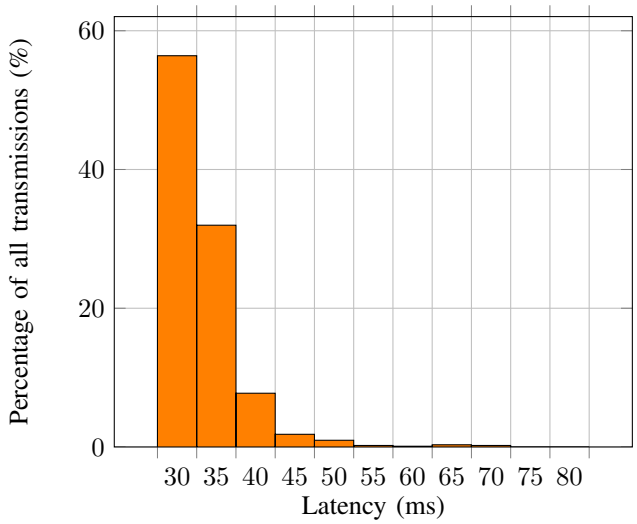


Fig. 9. Timing measurement: RTT including Internet latency between TCU on a leased line and TGU on a cloud-service provider

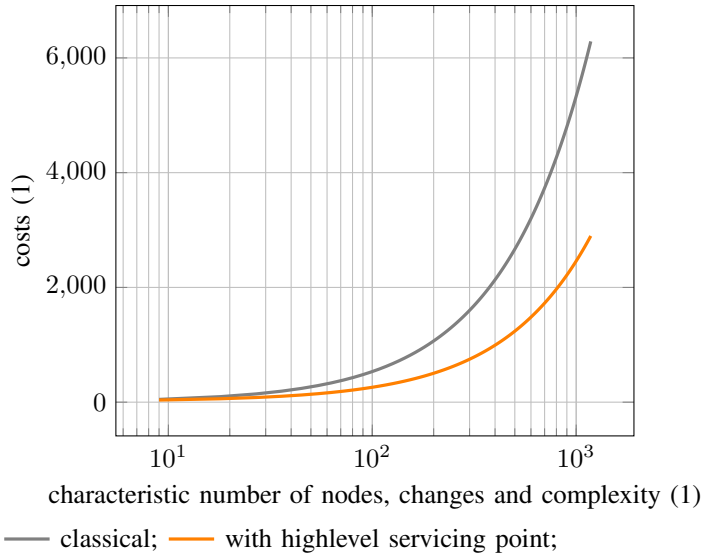


Fig. 10. Trend results of the economies of scale model with increasing number of changes, code complexity and time effort for each change

Figure 10). It shows that, according to the model, the proposed method brings significant lower costs, if the characteristic number of nodes, changes and complexity rises. The experimental implementation of the adapted multi-tier architecture shows, that a dynamic embedment of resource capacities and rich environment models is possible in a real application. A basic system operation is ensured, even during complete network fails, as experimentally proven. An incremental complement and improvement of sub-algorithms and model components becomes possible in a comfortable way, using Cloud-based resources on the Core-Peers as well as modern programming languages like Java as high-level servicing points.

VII. Conclusions and Further Work

In this article an adapted multi-tier architecture for future CPS was shown. A method for the dynamic embedment of resources and models from Core-peers into Edge-peers at runtime was described. This allows an effective scaling and a more adaptive behavior of automation systems in distributed systems. A basic system functionality is ensured, even when the network fails completely. Two algorithms for the dynamic embedment of resources were described. An application example was given from the field out of the renewable energies. The expected advantages of the described method were met. Timing measurements showed the practicle use of the proposed method. The economies of scale model showed the positive scaling effect for larger node numbers. This opens up a wide range of possibilities of advanced analysis and processing capabilities, triggered from lightweight Edge-peers. Further possibilities of dynamic resource sharing between CPS nodes are worth to explore.

Acknowledgment

This work is supported by the Baden-Württemberg Ministry of Science, Research and the Arts (MWK) within the scope of Cooperative Research Training Group.

References

[1] E. Lee, *Introduction into Embedded Systems*. UC Berkeley, 2011.

[2] J. Schlechtendahl, "Communication Mechanisms for Cloud based Machine Controls." Procedia CIRP Volume 17, Variety Management in Manufacturing — Proceedings of the 47th CIRP Conference on Manufacturing Systems, 2014.

[3] ——, "Study of network capability for cloud based control systems." 24th International Conference on FAIM, San Antonio, 20.-23.05.2014, 2014.

[4] M. Broy, *Cyber-Physical Systems (acatec DISKUTIERT)*. Springer, 2010.

[5] ——, "Cyber-Physical Systems - Innovationsmotor für Mobilität, Gesundheit, Energie und Produktion," Tech. Rep., 2011.

[6] S. Kowalewski, "Cyber-Physical Systems - A UMIC Perspective," RWTH AACHEN UNIVERSITY, Tech. Rep., 2015.

[7] B. Kölmel, *Cloud Computing und Eingebettete Systeme: System-Entwurf, Realisierung und Bewertung*. VDI/VDE Automation 2013, 14. Brachentreffen der Mess- und Automatisierungstechnik, 2013.

[8] J. Dell and T. Greiner, "Model-based platform design and evaluation of cloud-based cyber-physical systems (CCPS)," *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*, pp. 376–381, Jul. 2014.

[9] G. Engel, C. Stahl, M. Barth, T. Greiner, and D. Gorecky, "Cloud-basierte Automatisierung - Betrachtungen zur Verfügbarkeit," *atp edition*, vol. 3/2015, 03 2015.

[10] M. Engelsberger and T. Greiner, "Software architecture for cyber-physical control systems with flexible application of the software-as-a-service and on-premises model." IEEE, Mar. 2015, pp. 1544–1549.

[11] S. Kowalewski, B. Rumpe, and A. Stollenwerk, "Cyber-Physical Systems - eine Herausforderung an die Automatisierungstechnik?" Tech. Rep., 2012.

[12] M. Fowler, *Patterns of Enterprise Application Architecture*. Pearson Education, 2012.

[13] R. Buyya, *Cloud Computing: Principles and Paradigms*. John Wiley & Sons, 2010.

[14] Y.-K. R. Kwok, *Peer-to-Peer Computing: Applications, Architecture, Protocols, and Challenges*, ser. Chapman & Hall/CRC Computational Science, 2011.

[15] F. Konrad, *Planung Von Photovoltaik-Anlagen: Grundlagen und Projektierung*. Vieweg Verlag, Friedr, & Sohn Verlagsgesellschaft mbH, 2008.

CPU-based Covert- and Side-Channels in Cloud Ecosystems

Johann Betz, Dirk Westhoff
Hochschule Offenburg University
Offenburg, Germany
{*johann.betz, dirk.westhoff*}*@hs-offenburg.de*

***Abstract*—Covert and Side-Channels have been known for a long time due to their versatile forms of appearance. For nearly every technical improvement or change in technology, such channels have been (re-)created or known methods have been adapted. For example the introduction of hyperthreading technology has introduced new possibilities for covert communication between malicious processes because they can now share the arithmetic logical unit (ALU) as well as the L1 and L2 cache which enables establishing multiple covert channels. Even virtualization which is known for its isolation of multiple machines is prone to covert and side-channel attacks due to the sharing of resources. Therefore it is not surprising that cloud computing is not immune to this kind of attacks. Even more, cloud computing with multiple, possibly competing users or customers using the same shared resources may elevate the risk of unwanted communication. In such a setting the "air gap" between physical servers and networks disappears and only the means of isolation and virtual separation serve as a barrier between adversary and victim. In the work at hand we will provide a survey on weak spots an adversary trying to exfiltrate private data from target virtual machines could exploit in a cloud environment. We will evaluate the feasibility of example attacks and point out possible mitigation solutions if they exist.**

I. Introduction

While cloud computing is getting widely established, there are still many obstacles that lead to unused possibilities for both companies and end users. One of the most discussed factors when migrating to cloud based solutions is the security threat when entrusting a third party with sensitive data. The obvious risks in a cloud environment like untrustworthy providers, failure of service level agreement (SLA) fulfillment or location based security risks like e.g. storing the data in US, can be gauged and possibly be minimized beforehand. Despite such obvious risks, there are other security risks that are known for a long time but are still posing a threat since no general mitigation solutions have been found. Even more, the impact of threats like covert- and side channels does increase due to the fact that in a cloud environment multiple companies or end users share a multitude of resources and physical machines. In an Infrastructure-as-a-Service (IaaS) environment, competing parties can even share the same infrastructure and sources of virtual machine generation (image- and datastores). While virtualization stands for isolation, this is the case for known and protected ways of communication. Covert- and side-channels, on the contrary, can circumvent isolation and virtual machine (VM) boundaries. In addition to the lack of mitigation means, auditability is not easily carried out or even impossible in some cases for such channels. Therefore the nontransparent nature of many cloud providers and their services is even more risky because of unreliable audits with respect to serious security threats. Covert channels (CCs) use a shared resource or object to communicate between two colluding malicious processes by modifying content or properties of the resource. CCs are traditionally categorized as storage and timing channels (although for concrete channels classification is sometimes vague), as indicated from the Trusted Computer System Evaluation Criteria (TCSEC) in 1983 [1]. In storage channels a sender transmits bits by modifying resources or states of resources persistently (at least during a specific period of time) which is then detected by the receiver who infers the transmitted bits from the state. In timing channels the sender modifies the timing of a resource to enable the receiver to infer information (e.g. response or execution time by flooding with queries). Sender and receiver in the CC scenario collude and the sender is a malicious process placed in the target environment. Side-channels in contrast use only a receiver inferring information and data of the target by spying on the use of a shared resource. Therefore a designated sender is not needed and the target processes expose information silently. Both types of channels allow an attacker to infer sensitive data without any control by access mechanisms and audits. This represents a violation of confidentiality, privacy and auditability. The structure of the work is as follows: We will first discuss the feasibility of covert channels in section II. Section III will give an introduction to general aspects of mitigation. A survey on CCs and SCs in the "CPU" shared resource is then conducted in section IV-A and section V will conclude this paper.

II. Feasibility of Covert- and Side-channels

One important aspect of covert channels and side-channels is the feasibility discussion. However, it has to be carried out for every new infrastructure that changes access interfaces and shared resources. While for a side-channel to exist, only a badly programmed process or resource access component is needed, covert channels require a sender that is infiltrated into the target environment and a colluding receiver. In classic literature on covert channels, bandwidth is generally considered the factor describing the feasibility of the channel. How colluding processes are placed remains mostly undiscussed. While we do not want to cover this aspect to full extent we

can make some assumptions for a public cloud scenario in the IaaS service model. If considering such a cloud scenario, controlling the receiver can be regarded to be of few effort because the attacker can create an account/virtual machine which then contains the receiving process. The required co-residency with the desired target machine can also be forced in some cases or at least be detected (see [2]). The sender in contrast, has to be infiltrated into the target machine. Typically this is possible due to the means of distribution of trojans and viruses. In a cloud scenario this can be extended by some characteristics specific to a cloud.

- Updates / upgrades
- Cloud specific software obligated to be installed/used (e.g. inside a VM)
- Insecure datastores / storage
- Faulty image store / contaminated base images

Malicious updates or upgrades are also a problem outside of a cloud setting. Inside a cloud ecosystem they are under the control of the cloud provider or some subcontractor unknown to the victim. Therefore improper inspection of updates and upgrades easily enables them to contain the malicious process. Even if considering an infrastructure as a service scenario (where the customer has the most control), the cloud provider could oblige the user to use additional software which is then prone to malignant updates or yet contains malicious processes. Moreover, a cloud provider frequently does not host the underlying storage infrastructure. There might even be more than one subcontractor providing the requested capacities. Here the customer might not even know about such relaying. Therefore it is indeed possible that covert channel processes are introduced at the storage level (although at that stage data might also be directly exfiltrated which renders the CC useless). In a setting where users can create VMs on the fly, base images which are cloned or shadowed (e.g VMware Fast Provisioning[1]) are used. If there is a vulnerability in the image store, new VMs could be created with a malicious process yet in place. Additionally cloud app market places e.g. the MultiCloud Marketplace[2] allow third parties to create software or executable code e.g. RightScripts[3] that is then installed into a VM and in the worst case can contain malicious processes. Apart from bandwidth limitations or other obstacles related to each specific covert channel, we thus believe establishment of a usable CC in a public cloud to be feasible due to the aforementioned possibilities to deploy sender and receiver processes.

III. Mitigation Methods

Currently there exists no mitigation method which can be applied to any generic covert channel despite full isolation. However isolation is costly and in many cases not even possible to maintain. Nevertheless as with covert channels itself, mitigation means can be categorized. TCSEC recommends [1]:

- elimination of the channel
- limitation of bandwidth
- auditing covert communication

Elimination of the channel in most cases means avoiding resource sharing between competing parties by adapting the infrastructure or design of the system (e.g. by duplicating resources). An example for elimination in a cloud scenario would be separation of storage such that every VM gets its own storage. To avoid the possibility of constructing a CC regardless of separated hard disks, every connection to the storage, such as buses have to be separated and dedicated to one VM. Otherwise the access latency over the bus could again be used as a CC. A boundary (upper limit) for acceptability of a CC has been given by the TCSEC as 1 bit/second. Regarding this limit, it is possible to mitigate CCs by limiting the bandwidth of the channel to this boundary. Limitation can be achieved by introduction of noise or delay rendering the reception of data unstable or in the best case impossible. Noise itself can be produced by introducing randomization of shared resource access or extraneous processes randomly creating fake load. In most cases noise and delay affect the whole system by degrading performance and therefore represent a trade-off between security and performance. A solution that tries to minimize performance degradation while utilizing this kind of mitigation technique is the C^3-Scheduler proposed by the authors of the work at hand [3]. The suggested scheduler tries to mitigate a CPU cache covert channel by creating noise while trying to reduce performance degradation reusing processes identified as benign and already present in the system. If it is not possible to eliminate a CC or limit its bandwidth, it is desirable to at least audit (or detect) the covert communication over the channel to make it less usable for an adversary. Obviously, this method faces some difficulties. First of all a sufficient amount of data has to be collected which can be difficult also in a judicial way because auditing can raise new privacy concerns. If enough data (audit trail) has been collected, the CC has to be distinguished from legitimate communication / interaction with the shared resource and noise present in the system by default. In that regard another problem is the identification and distinction of sender and receiver. C2-Detector, a general framework for detecting CCs in the cloud has been proposed by Wu et al. [4]. C2-Detector uses a hidden markov model (HMM) and a bayesian detector to distinguish CCs from legitimate communication. Although applicable to multiple CCs, the CC has to be known and modeled into a configuration file beforehand. For side-channels the same basic ideas for mitigation, except auditing, can be adapted. Isolating an attacker from spying on the victims communication with some resource eliminates both possible CCs and side-channels. An example for a noise based mitigation technique focusing on cache based side-channels in cloud is ”Düppel” proposed by Zhang et al. [5]. Düppel changes the kernel of the guest OS

[1]https://pubs.vmware.com/vcd-51/topic/com.vmware.vcloud.admin.doc_51/GUID-4C232B62-4C95-44FF-AD8F-DA2588A5BACC.html

[2]http://www.rightscale.com/library

[3]Community created scripts for the RightScale Cloud, supposed to automatically configure a virtual machine - https://support.rightscale.com/12-Guides/Dashboard_Users_Guide/Design/RightScripts/

in a VM to periodically cleanse the cache used by its tenant. This introduces noise and obfuscates timing information an attacker could infer from spying on the cache. Because there is no active communication to the adversary, auditing side-channels is even more difficult than CC in which at least the possibility to recognize communication patterns exist.

IV. Shared Cloud Resources

Cloud systems in general require and base upon sharing resources to reduce costs both at the side of the cloud provider and at the customer side. One of the key criteria for cloud ”pay-per-use” where a customer only pays for the resources actually used is only possible due to dynamic resource allocation of shared resources. This holds for all service models offered e.g. Infrastructure as a service (IaaS), Platform as a service (PaaS) or Software as a Service (SaaS). Therefore it is possible to identify shared resources common to those service models:

- Memory architecture
- CPU including L2 and L3 Cache
- Storage
- Network
- Infrastructure
- Hypervisor

In regards to the mentioned service models, the easiest possibility for an attacker to exploit resource sharing for covert- and side-channels is IaaS which allows him to control a complete VM co-resident to the target VM. Therefore in the remainder of this work we suppose a cloud scenario or setting to be of the IaaS type. In the following subsection, we will review some of the proposed attacks and mitigation methods for the CPU architecture including its caches, since they can be regarded the most investigated shared resource up to now.

A. CPU based Covert- and Side-channels

A basic approach for creating a covert channel using the CPU exists by forcing changes in the CPU load to transfer information between sender and receiver. Okamura et. al. adapt this approach for cloud computing, evaluating the impact of virtual CPUs on the feasibility of such a channel [6]. Their ”Covert Channels using CPU loads between Virtual machines” (CCCV) achieves a bandwidth of 0.49 bit/s with 100 to 91% accuracy (ideal and non-ideal case). While such a channel is not difficult to establish, bandwidth and detectability minimize its feasibility. To eliminate the CC, the adversary or target VM has to be migrated to a different CPU after its detection. As the authors of CCCV outline, a chinese wall policy which can be used in the MAC-based security extension sHype [7] would also prevent VMs of competing users from executing on the same physical processor although ”competing” users would have to be identified beforehand. Noise could be easily introduced by extraneous processes generating fake load, however this probably introduces a noticeable performance degradation. Detection and auditing is possible due to the forced switching between high and low load which can be detected by pattern / anomaly recognition. Additionally, if a virtual machine often creates high load, the cloud system possibly solves this by migration to a less utilized host.

A covert channel using a shared component of the processors arithmetic logical unit (ALU) e.g. a multiplier in a simultaneous multithreading (SMT) environment applicable to cloud has been proposed by Wang et al. [8]. The sender repeatedly accesses the multiplier component of a core to send a 1-bit and executes multiple NOP instructions to send a 0-bit. The receiver can then measure the delay inflicted on the ALU operations. While this channel can achieve high bandwidths of approximately 500 kilobits/s it is questionable if such a SMT only CC can be regarded as a threat in common cloud settings since hyperthreading is disabled in most production systems (because of security reasons) and the ALU is therefore not shared. Elimination of the CC can be achieved by restricting VMs to different cores. Introducing noise or delays in the usage of a processors ALU is not an option because of the massive performance degradation. However, detection and audit of the channel is possible because of the irregularity in ALU component usage which might be observable.

An own category for shared resources could be addressed to cache based covert channels which have been extensively studied. L2 and L3 cache has already been targeted for CC and side channel construction. L1 cache is generally not taken into account because it is flushed upon every context switch. Ristenpart et. al. introduce two versions of a cache based covert channel [2]. In the simplest version, the sender is idling to transmit a ”0” and tries frantically to access the cache to transmit a ”1”. The receiver can then observe the latency when accessing a memory block. The refined version uses the *prime+probe* strategy proposed by Percival [9] (extended to *prime+trigger+probe*) and partitions the available cache lines into an even and odd set. The receiver fills every cache line with data (by accessing a memory block larger than the cache). To send a ”0” the sender then accesses every even cache line to evict the receivers data from that cache set. To transmit a ”1” the ”odd” cache set is used likewise. The receiver now probes the cache by accessing the even and odd cache sets subsequently while measuring the latency for each set, inferring the transmitted bit from the difference. While high bandwidth is achieved in a laboratory environment the channel itself was also tested under practical circumstances within the Amazon Elastic Compute Cloud (EC2) where noise and core migration reduce the bandwidth to 0.2 bit/s. Xu et. al. therefore further refine the channel by using a refined protocol and achieve a maximum bandwidth of 10.46 bit/s with 28.13% error rate and a minimum bandwidth of 1.27 bit/s with 0% error rate in an Amazon EC2 environment [10]. The proposed L2 cache channels suffer one important problem introduced by virtualization called *addressing uncertainty*. Virtual memory addresses are mapped to physical addresses by the hypervisor and therefore determination of the accessed cache line depends highly on the alignment of virtual and physical addresses and the mapping function used. Therefore the feasibility is

decreased and how to overcome *addressing uncertainty* is still an open research question. Elimination of the L2 covert channels could be achieved by the concept of oblivious RAM proposed by Goldreich and Ostrovski [11] which describes a method of access pattern hiding but for which a practical scheme has yet to be found. Noise can be introduced to limit the bandwidth or even render the L2 channels unusable by deploying techniques as the already mentioned "Düppel" extension proposed by Zhang et al. [5] or the scheduling method proposed by the authors of "C^3-Sched" [3]. Auditing could be possible by monitoring the cache miss count and pattern which possibly changes because of the frantic access and eviction of specific cache sets (resp. all cache lines) by the sender (resp. receiver).

A relatively new channel, "C5: Cross-Cores Cache Covert Channel" proposed by Clémentine et al. uses the L3 or last level cache (LLC) to construct a high bandwidth CC tackling the problem of addressing uncertainty and core migration [12]. C5 uses a protocol close to *prime+probe* and relies on the inclusive property of the LLC cache which means that the LLC is a superset of the L1 and L2 cache and every cache line that is evicted from the LLC gets removed from the L1 and L2 cache likewise. The receiver accesses (probes) its L1 cache and measures access time. If the sender wants to send a "1", it completely fills the LLC which in turn results in eviction of the receivers data from its L1 cache due to the inclusive property. Therefore the receiver measures "slow" access and infers a "1" because the data now has to be fetched from the main memory. Bandwidth of such communication was reported to be as high as 751 bits/s in a laboratory environment. However, evaluation in a production environment such as Amazon EC2 has not been carried out. Elimination of the channel can only be achieved by partitioning the LLC (e.g. an adapted StealthMem approach [13]) or removing the inclusive property (exclusive caches as already used by AMD) which would cause the obligation of a cache hierarchy redesign. Introduction of noise (e.g. by $C^3 - Sched$) or intermediary flushing of the cache (Düppel) would at least result in an increase of transmission errors (sender is idle but receiver infers 1 because its L1 cache is evicted by other processes uninvolved in covert communication). C5 could be detected by monitoring the LLC because fully evicting the LLC on a regular basis might be an anomaly in comparison to the cache usage without an active CC.

Side-channels using cache to gather information about other processes and particularly targeting cryptographic keys have been widely studied in non-virtualized environments. Ristenpart et. al. proposed their *prime+trigger+probe* mechanism to be used for cache utilization measurements which can then help inferring CPU load of a target machine [2]. They used the conducted load measurements to realize a coarse-grained side-channel attack called "keystroke timing attack"[4]. In an idle virtual machine where there is only the event of someone typing, a spike in the load can be brought together with a keystroke. While Ristenpart et. al. suppose that such a scenario is possible in practical settings as the Amazon EC2, feasibility and applicability of such a sophisticated attack is regarded very low. The attack itself however is a good demonstration of possible side-channel adaptations to the cloud. Mitigation methods include avoiding co-residency (elimination) as well as the mentioned possibilities for introduction of noise and delay (limitation of bandwidth). While auditing a side-channel is considered difficult, the repeated execution of the prime step (accessing the whole cache) can possibly enable detection and audit of the channel.

While Ristenparts side-channel targeted the coarse grained keystroke information categorically excluding cryptographic keys, the recent attack proposed by Zhang et. al. indeed targets such keys using a version of the *prime+probe* strategy combined with an analysis stage to infer code-paths taken in a cryptographic library (libgcrypt) [15]. In contrast to covert channels, in the *prime+probe* protocol used for side-channels, the receiver or spy fills the whole cache with its data (prime) and then identifies the cache lines evicted by the (unknowing) target through probing the cache. The information which cache lines have been evicted can then be used to classify cache patterns with a support vector machine (SVM). After noise reduction using a hidden markov model and code path reassembly, the initial code path that has been taken in the cryptographic library can be inferred and allows conclusion of ElGamal keys. Elimination of the proposed side-channel is possible by avoiding co-residency or implementing side-channel resistant algorithms. The first one might be impractical in a public cloud scenario although the possibility to execute sensitive operations as cryptographic computations in a dedicated instance. The latter only fixes a specific cryptographic library. Limitation of reliable reconstruction of keys by introducing delays and noise as well as detection and audit of such side-channels can be conducted as already mentioned for other *prime+probe* based channels.

Table I summarizes the covert channels based on CPU and Cache, their type (SC = side channel, CC = covert channel), bandwidth and categorization of mitigation means (E = Elimination, L = Limitation of Bandwidth, D = Detection, A = Audit).

V. Conclusion

We like to note that although this survey is by far not complete, we decided to publish such initial survey. In the work at hand, we could nevertheless show that while isolation in cloud environments is enforced through virtualization and mandatory access policies, it is still possible to construct various covert- and side-channels. Such channels circumvent

[4]The keystroke attack was first introduced by Song et. al. in a non-virtualized environment in 2001 [14].

Channel	Type	Bandwidth	Mitigation
CCCV	CC	0.49 bit/s	E, (L), D and A
ALU based	CC	500 kbit/s	E, D and A
L2 Cache based (Ristenpart)	CC	0.2 bit/s	E, L, D and A
L2 Cache based (Xu)	CC	10.46 bit/s	E, L, D and A
C5 L3 Cache based	CC	751 bit/s	E, L, D and A
Keystroke timing attack	SC	-	(E), L, D and A
Cryptographic key attack	SC	-	(E), L, D and A

Table I
CHANNELS MISUSING THE CPU AND CACHES

isolation by making use of one of the most important benefits of the cloud, namely shared resources. Elimination of such covert channels by separating resources is sometimes possible but often costly. Introduction of noise or delays can be used for bandwidth limitation of the presented channels but nearly always results in performance degradation of the system. Therefore auditing solutions for CCs and SCs should be integrated in the auditing frameworks already mandatory in some cloud services where possible. Despite all presented CCs and SCs, their feasibility in current production environments is still low because of either low bandwidth or difficulties in establishing the channel. Research for mitigation solutions however should not be abandoned because changes in technology and therefore in the cloud will always lead to new versatile attacks. Moreover as shown in this brief survey, there are almost no general mitigation means and specifically designed methods mostly decrease system performance. Also no guarantee for preventing data leakage through covert channels can be given by any of the analyzed countermeasures. Therefore currently adoption of such security extensions by cloud providers or developers of virtualization solutions for the cloud is highly questionable. Invention of new mitigation possibilities with low performance overhead, in the best case targeting whole cloud components, could lead to a wider acceptance and deployment in state-of-the-art clouds.

VI. ACKNOWLEDGMENT

The work of the authors was partly funded by the BMBF ProSeCCo[5] project in the program 'Forschung an Fachhochschulen'. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the ProSeCCo project or the BMBF.

REFERENCES

[1] "Department of defense trusted computer system evaluation criteria," *Computer Security Center Standard*, vol. CSC-STD-001-83, December 1983.

[2] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 199–212. [Online]. Available: http://doi.acm.org/10.1145/1653662.1653687

[3] J. Betz and D. Westhoff, "C3-sched - a cache covert channel robust cloud computing scheduler," in *Proceedings of the Internet Technology and Secured Transactions (ICITST), 2014*, Dec 2014, pp. 54–60.

[4] J. Wu, L. Ding, Y. Wu, N. Min-Allah, S. U. Khan, and Y. Wang, "C2detector: a covert channel detection framework in cloud computing," *Security and Communication Networks*, vol. 7, no. 3, pp. 544–557, 2014. [Online]. Available: http://dx.doi.org/10.1002/sec.754

[5] Y. Zhang and M. K. Reiter, "Düppel: Retrofitting commodity operating systems to mitigate cache side channels in the cloud," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13. New York, NY, USA: ACM, 2013, pp. 827–838. [Online]. Available: http://doi.acm.org/10.1145/2508859.2516741

[6] K. Okamura and Y. Oyama, "Load-based covert channels between xen virtual machines," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, ser. SAC '10. New York, NY, USA: ACM, 2010, pp. 173–180. [Online]. Available: http://doi.acm.org/10.1145/1774088.1774125

[7] R. Sailer, T. Jaeger, E. Valdez, R. Caceres, R. Perez, S. Berger, J. Griffin, and L. van Doorn, "Building a mac-based security architecture for the xen open-source hypervisor," in *Computer Security Applications Conference, 21st Annual*, Dec 2005, pp. 10 pp.–285.

[8] Z. Wang and R. B. Lee, "Covert and side channels due to processor architecture," in *In Proceedings of the 22 nd Annual Computer Security Applications Conference (ACSAC)*, Miami Beach, FL, USA, 2006.

[9] C. Percival, "Cache missing for fun and profit," in *In Proceedings of BSDCan 2005*, Ottawa, Canada, 2005.

[10] Y. Xu, M. Bailey, F. Jahanian, K. Joshi, M. Hiltunen, and R. Schlichting, "An exploration of l2 cache covert channels in virtualized environments," in *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, ser. CCSW '11. New York, NY, USA: ACM, 2011, pp. 29–40. [Online]. Available: http://doi.acm.org/10.1145/2046660.2046670

[11] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," *J. ACM*, vol. 43, no. 3, pp. 431–473, May 1996. [Online]. Available: http://doi.acm.org/10.1145/233551.233553

[12] C. Maurice, C. Neumann, O. Heen, and A. Francillon, "C5: Cross-cores cache covert channel," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, ser. Lecture Notes in Computer Science, M. Almgren, V. Gulisano, and F. Maggi, Eds. Springer International Publishing, 2015, vol. 9148, pp. 46–64. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-20550-2_3

[13] T. Kim, M. Peinado, and G. Mainar-Ruiz, "Stealthmem: System-level protection against cache-based side channel attacks in the cloud," in *Proceedings of the 21st USENIX Conference on Security Symposium*, ser. Security'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 11–11. [Online]. Available: http://dl.acm.org/citation.cfm?id=2362793.2362804

[14] D. X. Song, D. Wagner, and X. Tian, "Timing analysis of keystrokes and timing attacks on ssh," in *Proceedings of the 10th Conference on USENIX Security Symposium - Volume 10*, ser. SSYM'01. Berkeley, CA, USA: USENIX Association, 2001. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251327.1251352

[15] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-vm side channels and their use to extract private keys," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12. New York, NY, USA: ACM, 2012, pp. 305–316. [Online]. Available: http://doi.acm.org/10.1145/2382196.2382230

[5]ProSeCCo - Promotionsvorhaben zur Erarbeitung von Sicherheitserweiterungen für das Cloud Computing.

Pixel-wise Hybrid Image Registration on Wood Decors

M. Grunwald, J. Müller, M. Schall, P. Laube, G. Umlauf and M.O. Franz

Institute for Optical Systems, University of Applied Sciences Konstanz, Germany

***Abstract*—The detection of differences between images of a printed reference and a reprinted wood decor often requires an initial image registration step. Depending on the digitalization method, the reprint will be displaced and rotated with respect to the reference. The aim of registration is to match the images as precisely as possible. In our approach, images are first matched globally by extracting feature points from both images and finding corresponding point pairs using the RANSAC algorithm. From these correspondences, we compute a global projective transformation between both images. In order to get a pixel-wise registration, we train a learning machine on the point correspondences found by RANSAC. The learning algorithm (in our case Gaussian process regression) is used to nonlinearly interpolate between the feature points which results in a high precision mage registration method on wood decors.**

I. Introduction

Today a large proportion of all furniture or floors with a wood-like appearance are made of artificially printed wood decors instead of actual wood. Camera-based inspection is used to ensure that the printed decors do not differ from an initial reference. Based on the production environments, line scan cameras are often used for the digitalization of the printed decors. Many defect detection algorithms require an optimal registration of the print and the reference before comparison. This paper describes a registration method for images of wood decors with an accuracy of at least one pixel.

There are several difficulties for this registration process which partly result from the specific image structure of the wood decors and from the setup of the line scan camera. One problem is the repetitive structure of wood. This characteristic leads to a typical correspondence problem for this use case, i.e. image regions at different locations are erroneously matched due to their high visual similarity. Another characteristic of wood is that its visual structure is often oblong and thin—there are many edges but few corners to extract. Edges alone can only be used to locally match image regions in the direction perpendicular to the edge. Along the edge, no unique correspondence between image regions can be established. This is the aperture problem in optic flow processing [1].

Line scan cameras raise more difficulties. Either the line scan camera moves over the image that is digitalized or the printed decor moves on a transport system under the fixed line scan camera. Variations in the speed of movement cause partial stretching or compression of the image in both setups. The difficulty is that this type of image transformation will be different all over the image so that this has to be corrected locally. Similar local distortions arise also because of lens distortions in the line camera.

This paper addresses the problem of pixel-wise image registration on wood decors based on a hybrid registration method. The registration method is built in five steps. The first four steps are a part of the *global image registration* solution which includes the extraction of feature patches, the correlation of these patches, the calculation of model parameters and the validation of the quality. The global registration step can only correct for perspective transformations (including translations and rotations). The fifth step of the registration method can be considered as a *local registration* as local parts of the image are transformed differently to account for movement variations and lens distortion. This step is done with the help of a machine learning method, Gaussian process regression, which leads to a dense, pixel-wise correspondence between both images. This application of machine learning to image registration constitutes the novelty of this paper as this – to our knowledge – has not been done in the literature before.

The paper is organized as follows: we briefly discuss previous work in image registration on wood decors in Section II. In Section III, we present an overview of the global registration and describe the approach on local registration using Gaussian processes. A detailed description and experimental evaluation of each registration step are presented in Section VI. The paper concludes with a discussion in Section V.

II. Previous Work

Image registration strategies are divided into two main classes, both of which deal with different problems. The first class is global registration: an entire image is registered at once by finding its transformation parameters such as translation, rotation, scaling and shearing. In this case, every pixel of an image is transformed in the same way in order to match the other image [2]. The second class of registration problems is the registration of local image regions. In order to register these kinds of images, different transformations for different parts of the images are needed. Within these two classes, there are several subclasses which treat different registration problems [3]. Whereas there is a large literature on image registration in general, the specific problems arising in the registration of wood decors have not been addressed in detail.

In previous work, we addressed the problem of local image registration on wood decors using the classical Lucas-Kanade algorithm for optical flow [4]. This algorithm is widely used

to detect movements between pictures or within videos. The application of an optical flow algorithm to the registration process was also proposed by J.-P. Thirion [5]. If an optical flow algorithm is applied to two images which are not registered, it detects the transformations as independent movements of parts of the image. When applied to wood decors, the Lucas-Kanade algorithm ran into problems as the contrast of the wood decor image is not strong enough for the algorithm to detect the optical flow in all parts of the image. In these areas, no transformation could be predicted. Additionally, due to linear structure of the wood grain, the aperture problem affects the vast majority of all image regions so that in these regions only the flow component normal to the edges can be calculated. In order to solve these problems, we applied a variant of the famous *Horn and Schunck algorithm* [6] in which the diffusion of the flow vectors is weighted depending on a contrast-dependent confidence measure. While this method was capable of capturing the local variations caused by movement variations and lens distortion, it performed poorly on estimating the global transformation between the images due to the systematic underestimation of image displacements caused by propagating local displacement vectors affected by the aperture problem.

III. Registration approach

A. Feature selection

The geometric properties of the images are represented by extracted features. Two feature algorithms were examined and compared in this work. Features should have several qualities: (1) they need a strong invariance against small transformations; (2) they have to be localizable which means that the same features are found in both images and that it is possible to match these features; (3) it must be guaranteed that enough features can be found. In an ideal case, these features are evenly spread across the image.

The first feature algorithm examined was the Harris corner detector [7]. The advantage of the Harris corner detector is that it is easy to implement, efficient and that it finds corners independently of their orientation. A recursive Gaussian-like low pass filter was used as pre-smoothing method. For the calculation we used a four point central difference derivative operator. From these filters, the Harris detector computes a local "cornerness" function fo each pixel. A point is considered as a feature point when its "cornerness" exceeds a certain threshold. The threshold has to be adjusted according the image and the number of feature points needed. In order to achieve this, an algorithm was developed which is shown in Fig. 1. To obtain a faster convergence, the desired count of feature points is given in terms of an upper and a lower bound. The step width is only adjusted if the number of features is outside of these bounds.

Harris features have the disadvantage that they are only computed at one image scale and that the corners are found only with a maximum precision of one pixel. They also tend to be very sparsely distributed in the image. We therefore tested a second, scale invariant feature detector: the scale

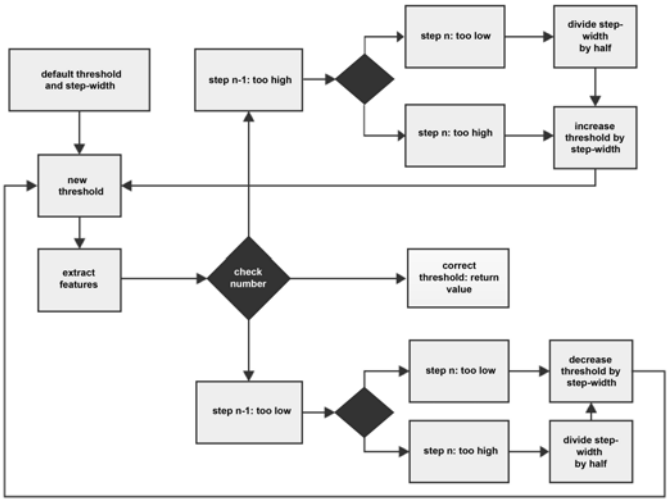


Fig. 1. Algorithm for determining the detection threshold of a feature. n is the number of iterations, the black rhombus symbolizes a decision based on counting the number of the extracted features.

invariant Laplace operator [8], referred to as blob features. The blob operator creates a Gaussian scale space, which is subsampled by a factor of two after every octave. Subsequently a Laplacian scale space is formed by subtracting adjacent layers of the Gaussian scale space. Blob-like image structures are identified as maxima or minima in the Laplacian scale space. The position of blob features can be calculated at sub-pixel accuracy. The underlying theory of the scale space and the blob detector can be found in T. Lindeberg, "Feature detection with automatic scale selection" [8].

B. Global registration

For the global registration we used the RANSAC (*Random Sample Consensus*) algorithm [9] based on the perspective transformation model [10]. The perspective transformation of an image is described by eight parameters a_i, b_i with $i = 1, 2, 3$ and c_j with $j = 1, 2$. That is why at least four corresponding control points in the reference and the transformed image are needed. For each of the point pairs (with index i), the image coordinates (x_i, y_i) in the reference are connected to (x'_i, y'_i) in the print by the equations

$$x'_i a_1 + y'_i a_2 + a_3 - x_i x'_i c_1 - x_i y'_i c_2 = x_i \qquad (1)$$

$$x'_i b_1 + y'_i b_2 + b_3 - y_i x'_i c_1 - y_i y'_i c_2 = y_i. \qquad (2)$$

These equations are linear in the unknown transformation parameters, so they can be solved by a standard least squares approach. We used the Moore-Penrose pseudoinverse for this purpose. However, the solution requires establishing point correspondences between both images which is the objective of RANSAC. Here, one chooses a large number of randomly chosen subsets of feature points in both images as candidate correspondences and tests the performance of the found transformation on other subsets of feature points. The best performing transformation is chosen to globally register the

images. This procedure converges very slow for the large number of features we detect in the images due to its stochastic nature. We therefore replace the random selection of subsets by a more directed form of selection: we extract a local image region around all feature points and find the most similar image region around a feature in the other image by searching for the candidate with the highest Pearson cross-correlation coefficient. All corresponding point pairs found in this way were ranked according to their cross-correlation. We restricted RANSAC to select its subsets only from the group of the highest ranking corresponding point pairs. This crucial step led to a considerable runtime improvement which made the proposed registration procedure feasible at all.

C. Local registration

After applying the global transformation to register both images we applied a second, local registration based on machine learning. The point correspondences between features were used as training data for a nonlinear regression technique. The two-dimensional image positions of the features in the transformed reprint were used as inputs and the x- or y-position of the correspondences in the reference were used as outputs. The result of the training is a dense mapping from 2d positions in the reprint to 2d positions in the reference which can visualized as a vector field (see Fig. 5). In other words, the machine learning interpolates the displacement field between both images at all pixels in the images, not only at the feature points and thus leads to a pixel-wise registration.

We choose Gaussian processes as our regression technique because they adapt well to non-linear functions with added noise, as described by Rasmussen and Williams [11]. This is made possible by the ability to use a covariance function in function-space. In addition, Gaussian processes include only a small number of hyperparameters that can be optimized using gradient descent.

Our implementation of the Gaussian process regression is based on kernel functions [12]. Kernel functions are used to calculate covariance measures in high dimensional spaces without actually transforming the input data. The choice of the kernel function influences the regression function and how well it fits the sample data. In our work, the shape of the mapping was unknown. We tested the Gaussian kernel which can model smooth displacement fields of arbitrary shape and the inhomogeneous polynomial kernel which restricts the shape of the displacement fields to follow two-dimensional polynomial curves. Details on the training of Gaussian processes can be found in the book by Rasmussen and Williams [11]. To find the hyperparameters of the Gaussian process, we used a gradient descent scheme on a smoothed form of leave-one-out error on the training set (Geissers surrogate predictive log probability) [11].

IV. Experimental results

To demonstrate the effectiveness of the proposed pixel-wise hybrid image registration approach we applied the method on different test sets with known and unknown transformations. In order to get representative results the cross validation method was used. For each test set the feature points were divided into equal random subsets and all but one of them were used for the calculation of the solution. This solution was afterwards applied to the unused subset and the error rate of the result was calculated. The error was measured in terms of the average absolute value of pixel deviation between the transformed and true feature point in the test set.

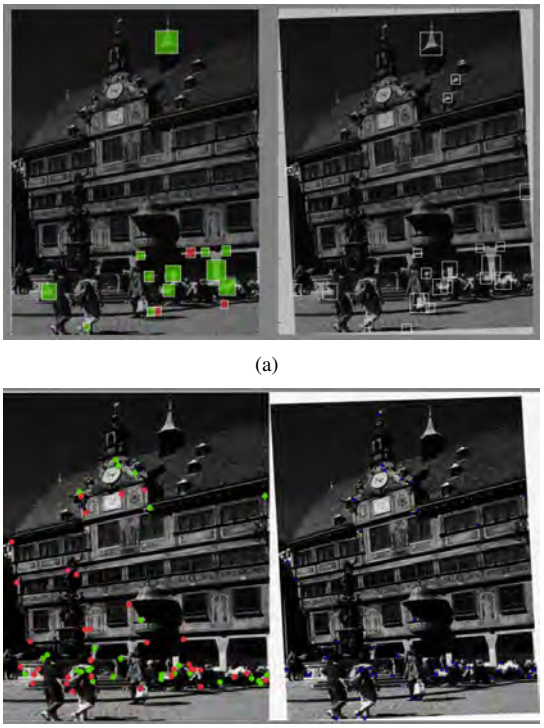
The test sets were divided into different categories. The first test category was artificially generated by applying known transformations. The town hall of the city of Tübingen and an artificially created wood decor served as test images. The contrast to the natural scene is used as an example to demonstrate the specific characteristics of wood decors. The second test category was based on scans of wood decors. Multiple scans of the same decor were made with different scan positions using a line scan camera.

A. Feature extraction

Since feature extraction is important for both the global and the local registration, we first examined the quality of the feature extractors. The term quality is defined as a number of properties of the feature extraction algorithm. The most important property is that the algorithm finds the same features in both images. Thus, it must be resistant to the noise of the camera and the transformations that were applied to the images. The number of features that can be extracted from the image is also of importance. At least four matched features were necessary for the calculation of the perspective transformation. For increasing robustness against imprecise features, ten feature points were used to calculate the perspective transformation by using the pseudo inverse. Additionally, five more features were needed to verify the precision of the calculated parameters. $1/6$ of the features could not be used as they were needed for the cross validation of the whole system. Alltogether, at least about 20 correct matched features were needed for our solution. The precision of the features is a further measure of quality. To achieve the objective of a pixel-wise registration of images, the features need to have a precision of at least one pixel.

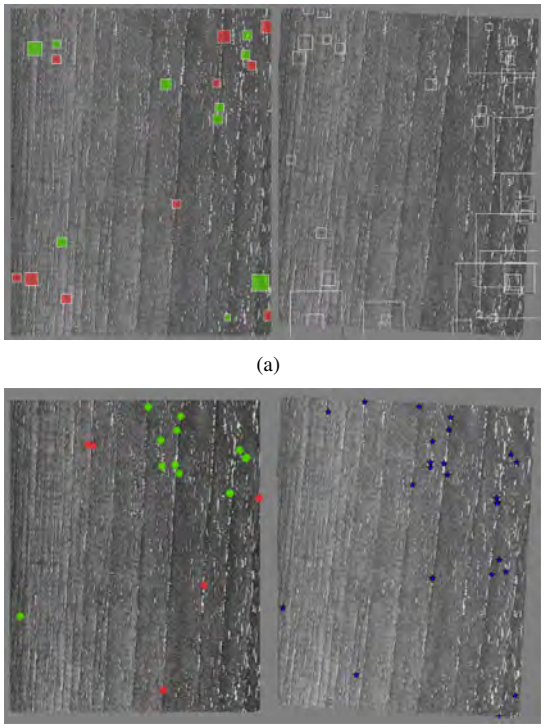
The first test was conducted to compare the feature extraction on natural scenes and wood decors. Harris and blob features were extracted from the test images and matched manually to ensure that no mismatch was produced by the correlation algorithm (see Fig. 2). 12 out of 15 features were extracted by the blob algorithm from both images which results in a retrieval ratio of 80%. The Harris algorithm extracted 29 out of 63 from both images, resulting in a retrieval ratio of a about 46%.

In the second image (artificial wood decor, see Fig. 3), the blob algorithm extracted 10 out of 20 features from both images (50% retrieval rate), the Harris algorithm 12 out of 17 features (70% retrieval rate). Obviously, the quality of the feature detector strongly depends on the texture of the images. Blob features are optimized for images with natural scenes such as landscapes or buildings which is shown by



(a) (b)

Fig. 2. The left images are the originals; in the right images, translation and rotation are applied to the image. The features marked in green can be retrieved in the second image. The red features could only be found in the first image. (a) Correlated blob-features of the town hall of Tübingen. (b) Correlated Harris-features of the town hall of Tübingen.



(a) (b)

Fig. 3. Same as Fig. 2 for the wood decor.

TABLE I
RESULTS OF THE FEATURE ALGORITHMS ON SCANNED WOOD DECOR.

	Inliers / outliers	**Precision**
Harris, wood1, T1	2.76	0.38
Blob, wood1, T1	1.22	0.18
Harris, wood1, T2	2.15	0.40
Blob, wood1, T2	0.97	0.20
Harris, wood2, T1	1.49	0.28
Blob, wood2, T1	1.06	0.18

their higher retrieval rate for this case, whereas the Harris extractor seems to be more suitable for wood decors. We also measured the average cross-correlation coefficient for both feature algorithms which was much higher for blob features as compared to the Harris features in both images.

In a second experiment, a full registration and a calculation of the ratio of outliers (incorrect correspondences) to inliers was conducted on the scans of the wood decors where the correct transformation is unknown. The results shown in Table I can be seen as indicators of two aspects of quality: the number of common features retrieved from both images and the quality or distinctiveness of the image areas surrounding the feature points.

Table I is structured as follows: the first column describes the test setup: the algorithm, the decor and the transformation used. The number after the transformation in *wood1* indicates two different positions on the same decor during the scanning. The second column contains the quotient of the number of matches (correct matches with a distance smaller than one pixel) divided by the number of mismatched features. As can be seen, the Harris detector always found a larger number of correctly matched features than the blob detector. The third column shows the mean distance of all inliers. This value should be low in order to obtain a high precision. The values of the third column show that the blob features are more precise than the Harris features with a relative improvement of at least 100%. The last aspect of quality is that the algorithm has to be able to extract a sufficient number of features from the image. As we said in the beginning of this section, at least 20 correctly matched features are needed for finding a reliable solution. To guarantee that this can be achieved we found in our experiments that at least 150 feature points have to be extracted from both images. To control the number of features, the threshold for the minimum contrast of a feature was adjusted according to the algorithm described in Fig. 1.

B. Image registration using Gaussian processes

Since this was the first time that Gaussian processes are applied to image registration we first tested whether this method is capable of estimating a known displacement field of a complex shape. Due to its clear horizontal and vertical edges the image of the town hall of Tübingen was chosen for this test. The town hall image was warped two times on a grid, shown in Fig. 4a. No global transformation was applied. The calculation of the Gaussian process response was done by

TABLE II
PRECISION OF THE GAUSSIAN PROCESSES CORRECTION WITH A GAUSSIAN KERNEL.

	[1]without GP	[1]with GP	[2]F-score
Wood1, T1	0.1869	0.1891	3.17
Wood1, T2	0.2189	0.1006	1.17
Wood2, T1	0.1949	0.1210	3.00

[1] Average precision in 5-fold cross-validation.
[2] Number of improved features divided by degraded features.

TABLE III
PRECISION OF THE GAUSSIAN PROCESS CORRECTION WITH AN INHOMOGENEOUS POLYNOMIAL KERNEL.

	[1]without GP	[1]with GP	[2]F-score
Wood1, T1	0.1755	0.1800	2.60
Wood1, T2	0.2259	0.2044	2.07
Wood2, T1	0.1700	0.1570	5.66

[1] Average precision in 5-fold cross-validation.
[2] Number of improved features divided by degraded features.

a set of correlated blob features as trainings points. For this configuration a Gaussian kernel was used.

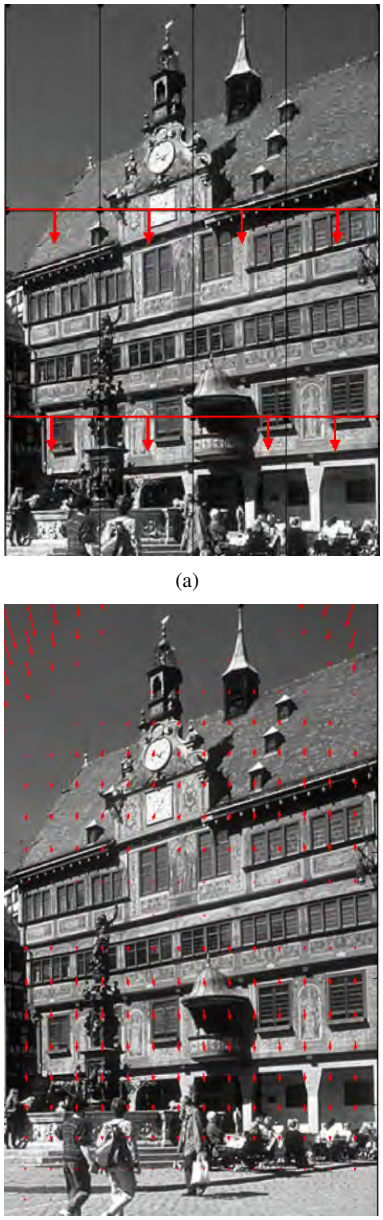
The result of the Gaussian process can be seen in Fig. 4b. The first grid line of the warp in the Gaussian response is where the arrows change the direction. The second warp was done in the same direction as the first line. The resulting arrows, which point in the same direction, follow the correct transformation. The long arrows in the sky above the town hall indicate an incorrectly learned transformation in this region. As the contrast was very low in this area, no features for the training the Gaussian process could be extracted. As a consequence, the predicted displacements vary widly in this area.

C. Hybrid image registration

These tests analyze the performance of the full hybrid image registration method on the scanned wood decor images. First the global registration method were applied to the feature points. In the second step, these corrected features were used to train the Gaussian processes. As a last step, the Gaussian processes were used to predict a correction of unseen feature points in a validation set.

Fig. 5a shows the prediction of the Gaussian processes. The error we observed in the previous experiment in the sky region of Fig. 4b also occurred in the corners in Fig. 5a, where due to the low contrast along the image edges no feature points could be detected which led to a high uncertainty in the prediction of the Gaussian process. The zoomed box shows an enlarged portion of the displacement field. The results for the inhomogeneous polynomial kernel are shown in Fig. 5b. The edge effect observed in the Gaussian kernel is much less pronounced here.

A quantitative evaluation is shown in Table II and III, indicating a significant improvement in accuracy for most cases.



(a)

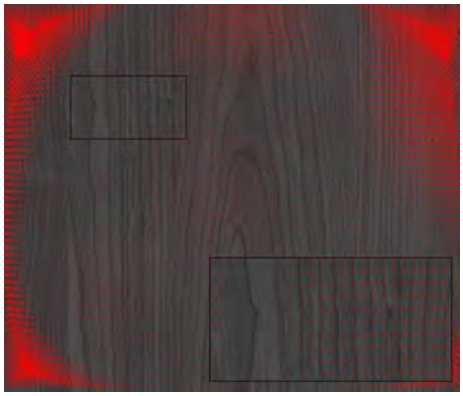
(b)

Fig. 4. Image (a): grid warp of the image. Image (b): response of the Gaussian process.

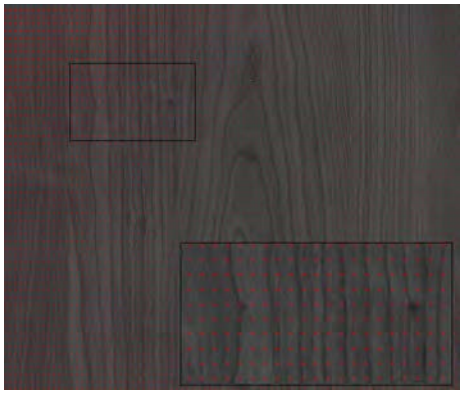
V. DISCUSSION

We presented a method for pixel-wise hybrid image registration on wood decors. Our experiments have shown a considerable improvement in the registration quality using Gaussian processes for local registration.

The results of several experiments show that the blob features achieve a higher accuracy. As a first step, the extracted blob features are correlated by using the Pearson cross-correlation coefficient. To guarantee that the succeeding calculations can be done without outliers, only the feature pairs with the highest correlation are retained. These feature pairs are used as input values for the calculation of the



(a)



(b)

Fig. 5. (a) Prediction of a Gaussian process with a Gaussian kernel on wood decor; (b) prediction of a Gaussian process with an inhomogeneous polynomial kernel on wood decor. Note that the arrows of the displacement field are not to scale, but enlarged for better visibility.

transformation parameters of the projective transformation model using the RANSAC algorithm. To increase the precision of the registration, the local registration is used to interpolate between the already globally corrected feature pairs. Thus, the features are used as training data for the Gaussian processes.

The results of the global and the local registration process depend both on the precision and reliability of the extracted features. The Harris algorithm has better values with regard to the ratio of inliers to outliers, but worse values in terms of the position precision of the features. Since the blob algorithm still has enough inlier features, its higher precision turns out to be more important since the registration should be as precise as possible. For this reason, blob features appear more suitable for registering wood decors according to our experiments. The global registration appears to be very robust against imprecise features or outliers since only the subset with the highest confidence is used to calculate the transformation.

We found a significant improvement in precision due to local registration with Gaussian processes for most cases. In the first test case, however, it is possible that the global registration found a very good solution for the transformation, so that the correction by the Gaussian processes did not improve the precision of the features. The second and third tests results in table II show that the precision is improved – more than 100% in the second test. It comes as a bit of a surprise that the quotient of the second test is the lowest. This might result from the fact that the features which are corrected are rather imprecise after global registration and the correction of those seems to have a strong impact on precision.

The results of the Gaussian kernel and the inhomogeneous kernel are comparable. The predicted correction in both cases is almost completely smooth. It seems that the predictions of the Gaussian kernel vary to higher degree across local image regions and that it is more susceptible to an inhomogeneous feature distribution, whereas the inhomogeneous polynomial kernel generally gives a very smooth prediction. The training data tended to be similar but not equal. The high value of the F-score in the last test in Table III shows that there was still a systematic error in the global registration which the inhomogeneous polynomial kernel corrected best.

As a consequence, in order to use a Gaussian process with a Gaussian kernel, the features should be evenly distributed across the image. There are several options to deal with the areas with a low number of features. One option is to use the predicted variance of the prediction of the Gaussian process which can be calculated together with the predictions [11]. In a low contrast area, the predicted variance would be quite high. In these areas, a correction with a Gaussian process can be disabled or the correction can be done by using the predictions from neighbouring features.

The overall algorithm has not been optimized for real-time application yet. Especially the training process is computationally expensive and needs to be improved to be useful in production systems, e.g. in an inspection system for printed wood decors.

REFERENCES

[1] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International journal of computer vision*, vol. 12, no. 1, pp. 43–77, 1994.

[2] M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup, "Efficient subpixel image registration algorithms," *Optics letters*, vol. 33, no. 2, pp. 156–158, 2008.

[3] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and vision computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[4] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision." *IJCAI*, vol. 81, pp. 674–679, 1981.

[5] J.-P. Thirion, "Image matching as a diffusion process: an analogy with maxwell's demons," *Medical Image Analysis*, 1998.

[6] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[7] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Citeseer, 1988, p. 50.

[8] T. Lindeberg, "Feature detection with automatic scale selection," *Int. J. Comput. Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[9] M. Brown and D. G. Lowe, "Recognising panoramas," *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 1218–1225 vol.2, 2003.

[10] R. Szeliski, *Image formation*, ser. Texts in Computer Science. Springer London, 2011.

[11] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006.

[12] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2001.

The overview of Public Key Infrastructure based security approaches for vehicular communications

Artem Yushev, Axel Sikora
Institute of Embedded Systems and
Communication Electronics
Offenburg University of Applied Sciences
Offenburg, Germany 77652
Email: {artem.yushev, axel sikora}@hs-offenburg.de

***Abstract*—Modern transport infrastructure becomes a full member of globally connected network. Leading vehicle manufacturers have already triggered development process, output of which will open a new horizon of possibilities for consumers and developers by providing a new communication entity - a car, thus enabling Car2X communications. Nevertheless some of available systems already provide certain possibilities for vehicles to communicate, most of them are considered not sufficiently secured. During last 15 years a number of big research projects funded by European Union and USA governments were started and concluded after which a set of standards were published prescribing a common architecture for Car2X and vehicles on-board communications.**

This work concentrates on combining inner and outer vehicular communications together with a use of Public Key Infrastructure (PKI).

Keywords*—Automotive engineering, Embedded Software, TLS, Security, Vehicle safety.

I. Introduction

Traditionally car manufactures are concerned with passenger safety in every possible situation on a road or parking lot, human life is treated as most important thing. Several decades ago car stopped being only a simple transport, but begun being involved in everyday live of a driver and passengers, car's on-board computer started to provide information about traffic and navigation. Together with media options they formed an infotainment domain, and more than 15 years ago a new concept of Car2X communications was born and safety concept was expanded on security domain as well, where "X" stands for following: an infrastructure (signs, traffic lights, wireless roadside units), pedestrians, and a car. Moreover, about seven years ago a car was believed to be a closed system, along with internal electronics evolution became an object of attacks which can lead to a human injure. Even though security objectives for Car2X and on-board communications differ from traditional, it is believed that generic approach still can be applied, for example asymmetric cryptography, also known as Public-key cryptography.

This paper provides an overview of research and standardization activities, as well as proposes a common approach for security in the context of in-Car and Car2X communications under Public-key cryptography domain. It is structured as follows: Chapter II provides a brief overview of available projects results. The analysis of available standards and related projects is outlined in Chapter III and IV, respectively. Chapter V present the authors view on a unified architecture which possibly can solve a problem highlighted previously. Finally Chapter VI summarizes the paper and provides an outlook to future work.

II. Overview

Asymmetric cryptography certainly becomes a de facto standard for systems where entities can't fully trust each other, because it provides a possibility to encrypt messages and verify digital signature for public key holder peer, as well as decrypt messages and generate signature for a private key holder. This obviously holds true for vehicular communications in various use cases, for instance technical report from ETSI [1] defines ten application areas for Car2X communications:

- Stationary vehicle warning accident vehicle problem
- Traffic condition warning (includes traffic jam ahead warning)
- Signal violation warning (includes stop sign violation)
- Road work warning
- Collision risk warning from Really Simple Syndication (RSS)
- Decentralized floating car data precipitations, road adhesion, visibility, wind
- Regulatory/contextual speed limits
- Traffic information & recommended itinerary
- Limited access, detour notification
- In-vehicle signage

In fact security objectives for cases above are general, namely: Confidentiality, Integrity, Availability, Accountability,and Authenticity [2]. Despite the fact that some sub-objectives seem to contradict each other; e.g., only authorized users should be able to participate in the system, on the other hand privacy aspects require anonymity or non-traceability, the main cryptographic challenge in Car2X networks is to protect many relatively short messages, which have to be authenticated, i.e., protect integrity of sender and of data, in a mobile Ad Hoc network. The communication happens in broadcast or unicast mode, and latencies are very critical [3].

Slightly different situation is with on-board communications, where integrity and authenticity requirements come to

the fore, where every Electronic Control Unit (ECU) can be considered as potentially malicious device. This holds true especially for modern cars which comprised of hundreds ECU's and provide various wireless and wired interfaces.

Nevertheless security objectives for distinct communication areas are different and international standardization committees established separate working groups, authors believe that two areas can be secured in a frame of common and unified Public Key Infrastructure [4].

III. Standardization activities

A. IEEE 1609.2

The IEEE 1609 working group has developed the IEEE 1609 Family of Standards for Wireless Access in Vehicular Environment (WAVE) USA [5], which defines an architecture and a complementary, standardized set of services and interfaces to secure car-to-car (C2C) and car-to-infrastructure (C2I) wireless communications. The IEEE 1609.2 standard specifies a set of management plane WAVE Security Services available to applications and processes running in the various layers of the WAVE protocol stack as shown in Figure 1. The WAVE Security Services consist of:

- *Security processing services:* provide processing that is performed to enable secure communications that comprise secure data and secure WAVE Service Advertisements (WSAs);
- *Security management services:* provide certificate management services that are provided by the Certificate Management Entity (CME) and that manage information related to the validity of all certificates; provide security management services that are provided by the Provider Service Security Management Entity (PSSME) and that manage information related to certificates and private keys that are used to send secured WSAs.

The services and entities within the WAVE Security Services are shown in Figure 1, which also shows Service Access Points (SAPs) that support communications between WAVE Security Services entities and other entities. This standard specifies the security processing via primitives defined at these SAPs. Horizontal boundaries within the WAVE Security Services in Figure 1 are abstract and do not correspond to horizontal boundaries within the Data Plane. Data passed across the SAPs in this standard is assumed to be secure and trustworthy. This standard does not provide mechanisms to ensure the trustworthiness of this data.

B. ETSI ITS

Core European activities in this field take place at European Telecommunications Standards Institute (ETSI) as European regulatory body, which received the official mandate to standardize C2X. ETSI's Technical Committee on Intelligent Transport Systems (TC ITS) is responsible for the production and maintenance of standards to support the development and implementation of ITS communications and services across the network, for transport networks, vehicles and transport users. Nevertheless ETSI ITS standards were composed in tight cooperation with Car-to-Car Communication Consortium

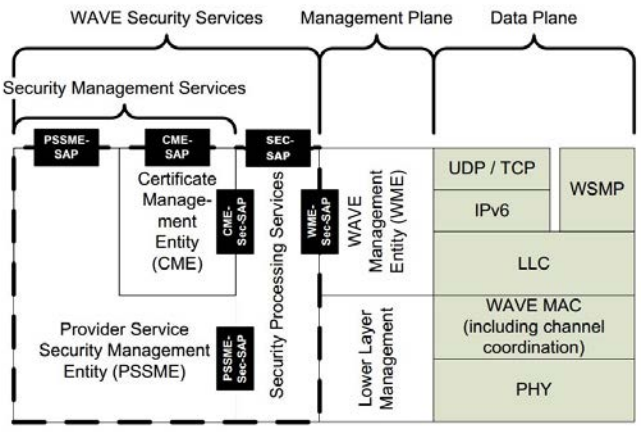


Fig. 1. WAVE Protocol stack for IEEE 1609-2013 in USA [5]

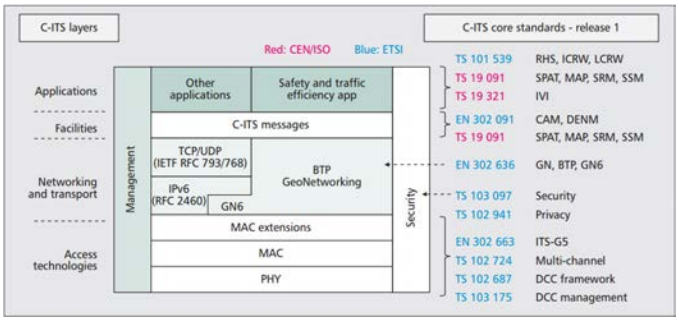


Fig. 2. Protocol stack and Release 1 core standards for C-ITS in Europe [6]

(C2C CC) and in most aspects resemble IEEE 1609 standards. There are some additional points which were added to support better security. Comparing to IEEE 1609 WG ETSI WG5 explicitly assigns security services for different application groups. But security primitives are the same: Authentication and Authorization, Confidentiality and Privacy. Figure 2 shows security as a vertical layer adjacent to each of the ITS layers but, in fact, security services are provided on a layer-by-layer basis so that the security layer can be considered to be subdivided into the four basic ITS processing layers as shown in [7]. According to ETSI TS 102 941 [8], the ITS-Station (ITS-S) security life-cycle includes four stages: manufacture, enrollment, authorization,and maintenance.

Communications security services [8] require, by definition, more than one element within their functional model. The principle functional elements and reference points between them can be determined by considering a simple ITS communications scenario such as that shown in Figure 3 [9]. This shows an ITS-enabled vehicle which needs to communicate with the following entities: an enrollment authority (lines 1 and 2), an authorization authority (lines 3,4,5), other ITS-equipped vehicles (line 6).

C. Car-to-Car Communication Consortium (C2C-CC)

Being a non-profit, industry driven organization initiated by European vehicle manufacturers, C2C-CC became one of the main contributors to an ETSI standardization body. The main document regarding security aspects is the C2C-CC PKI Memo distributed only within members of an alliance,

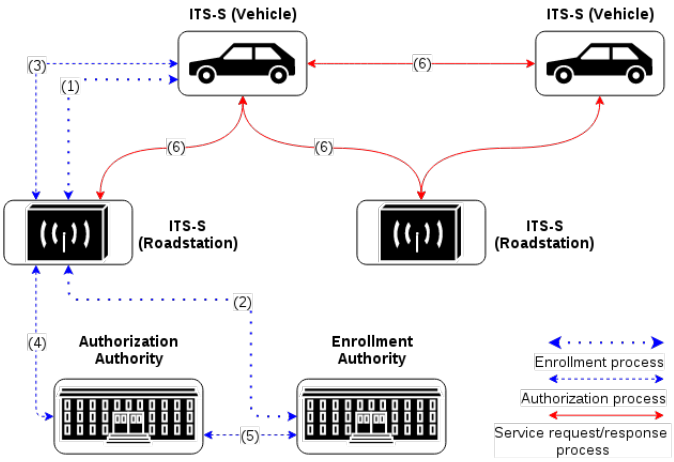


Fig. 3. The placement of security services within the ETSI ITS station architecture

however different sources can help to understand the general architecture and approach to reach required security level. Moreover C2C-CC security architecture is used as a base for PRESERVE security architecture, which will be explained further.

IV. Implementation Activities

A. Digital Tachograph Systems (DTS) project

The digital tachograph succeeded the analogue tachograph as a result of European Union regulation 1360/2002 that made digital tachographs mandatory for all relevant vehicles manufactured after August 1, 2005. Digital tachographs are required as of May 1, 2006 for all new vehicles to which EWG regulation VO(EWG)3820/85 applies, as is published in the official newsletter of the European Union L102 from April 11, 2006.

A digital tachograph system consists of a sender unit mounted to the vehicle gearbox, the tachograph head and a digital driver card. The sender unit produces electronic pulses as the gearbox output shaft turns. These pulses are interpreted as speed data by the head.

The asymmetric cryptographic system based on the standard Public Key Infrastructure (PKI) is used for securing the communication between the Vehicle Unit (VU) and the tachograph card. The DTS European Root Policy [10] defines the general conditions for the PKI concerned and accordingly contains more detailed information.

According to [11] DTS is well working system regulated by German government, which is going to be a part of global ITS. However DTS project shows an application area where Car2X domain is tightly bound with on-board communications.

B. PRESERVE project

PRESERVE (Preparing Secure Vehicle-to-X Communication Systems) is a concluded EU project with a total budget of 5.438 M€ and 395 man months resources.

Three workshops were organized by this project aimed on solving harmonization and implementation issues combining efforts and outcome of different successful projects. A distinctive feature of the PRESERVE project is that it focuses

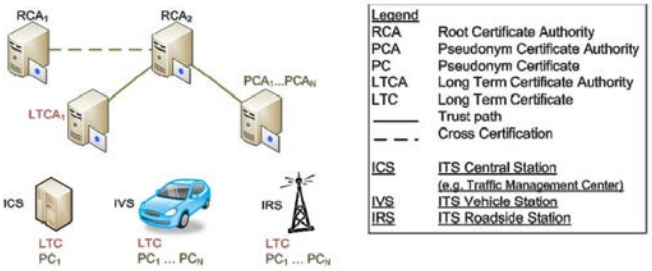


Fig. 4. C2C-CC System Architecture [11]

not only on the PKI part but also on secure communication for on-board Equipment to be compliant with ETSI standards complementing them with necessary implementation details.

The Vehicle Security Architecture (VSA) [12] is designed to protect on the one hand the V2X communication between ITS stations and the on-board communication system by using results and solutions from the previous projects SeVeCom [13] [9] and EVITA [14], [15]. On the other hand, privacy protection is a very important aspect in Vehicle-to-X (V2X) communications and therefore relevant mechanisms are considered in related processes. They are based primarily on results of the projects SeVeCom and PRECIOSA [16]. Furthermore, the VSA aims at being compatible with specifications defined by standardization bodies such as the ETSI, the IEEE, and the industrial driven consortium C2C-CC. In order to be practically relevant for close-to-market Field Operational Tests (FOTs), operational and evolutionary aspects such as re-usability, adaptability, scalability, and cost-effectiveness are considered by the VSA. Based on this VSA a V2X Security Subsystem (VSS) is created that combines different and partially enhanced security and privacy mechanisms from previous projects. The PRESERVE VSS aims to be usable in future V2X communication system implementations. The main document regarding security aspects is the C2C-CC PKI Memo distributed only within members of an alliance.

PRESERVE VSA consists of three different basic types of Certification Authorities (relationships depicted in Figure 4): Root certificate authority (RCA), Long-Term certificate authority (LTCA), Pseudonym certificate authority (PCA). The role of the Root CA is to define common policies among all subordinate certificate issuers. The RCA only issues certificates for Long-Term CAs and Pseudonym CAs, which are valid over long periods. A certification process which needs interaction with the RCA is only required once a new LTCA or PCA is created, and when the lifetime of an LTCA or PCA certificate expires. If there are multiple RCAs, they may cross-certify each other. For mutual trust between Root CAs, a cross certification is reasonable because it allows more flexible trust relationships between the Root CAs. However, the overall number of RCAs shall be kept as small as possible.

In the PRESERVE solution as shown in Figure 4, a Root Certificate Authority (depicted in Figure 4 as RCA2) on a European level is proposed to be used as central trust anchor. In order to extend the solution for world-wide interoperability, all existing RCAs may cross certify each other. Every cross certification is done with two new certificates stating the mutual trust status between both Root CAs. Any other cross certification between CAs other than RCAs, i.e. between Long-

term CAs and between Pseudonym CAs, is not allowed. Every LTCA has a Long-Term CA certificate that is signed with the private key of the Root CA. With a similar process the Root CA issues Pseudonym CAs that provide valid Pseudonym CA certificates afterwards. In the proposed design only the Root CA is able to issue Long-Term and Pseudonym CAs. Afterwards, the LTCA issues for each responsible ITS station one Long-Term Certificates (LTC), that is valid for a longer period. Each LTC created by a LTCA is dedicated to identify and authenticate the respective ITS station within the PKI and potentially other services, but they are never exposed to the ETSI ITS 5GHz based wireless (ETSI G5A) communication for privacy reasons. In contrast, Pseudonym Certificates (PCs) issued by PCAs are used for the ETSI G5A broadcast communication. PCs are designed to have a short lifetime and have to be exchanged frequently. The VSS is placed inside the ITS station and aims to provide necessary security services as defined by ETSI in the ITS station reference architecture [24]. The VSS is connected to the on-board networks, the application unit that runs the V2X applications and the V2X communication entity that is connected with the outside world (i.e. ETSI ITS-G5A network).

C. AUTOSAR framework

AUTOSAR (AUTomotive Open System ARchitecture) [17] is an open and standardized automotive software architecture (for in car communications), jointly developed by automobile manufacturers, suppliers and tool developers. It is a partnership of automotive Original Equipment Manufacturers (OEMs), suppliers and tool vendors whose objective is to create and establish open standards for automotive E/E (Electrics/Electronics) architectures that will provide a basic infrastructure to assist with developing vehicular software, user interfaces and management for all application domains.

Nine Core members include the BMW Group, Daimler AG, Ford Motor Company, General Motors, Opel, Toyota Motor Corporation, PSA Peugeot Citron, Volkswagen and automotive suppliers Bosch, Continental AG and Siemens VDO (now Continental AG).

Nowadays AUTOSAR architecture is supported by all big car and ECU manufactures, moreover according to a reviewed literature ITS infrastructure is going to be implemented within AUTOSAR specification. Due to AUTOSAR framework prevalence for car manufacturer it is reasonable to assume that all security features will be implemented inside this framework. In fact this environment has already standardize crypto abstractions and security management entities, although limited in functionality [18].

V. Approach

A. Basic idea

The basic idea of considered approach was to review ongoing and concluded security related activities–briefly discussed in previous chapter–from the perspective of unified and harmonized Car2X and in-Car security systems (see Figure 5), particularly take as a base C2C-CC architecture and combine it with standardized methods for on-board communications. Moreover a special attention was paid for in-car communications, where most of links are established via CAN bus,

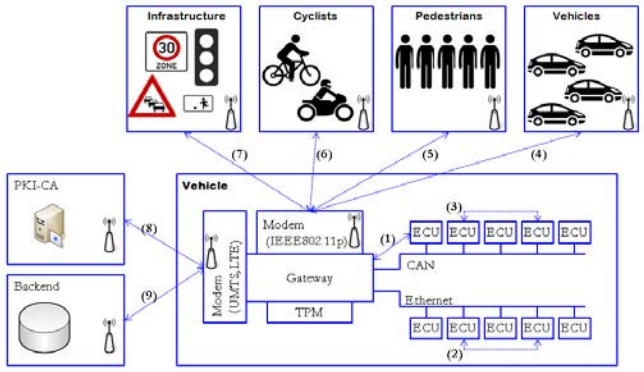


Fig. 5. Architecture of the anticipated communication system

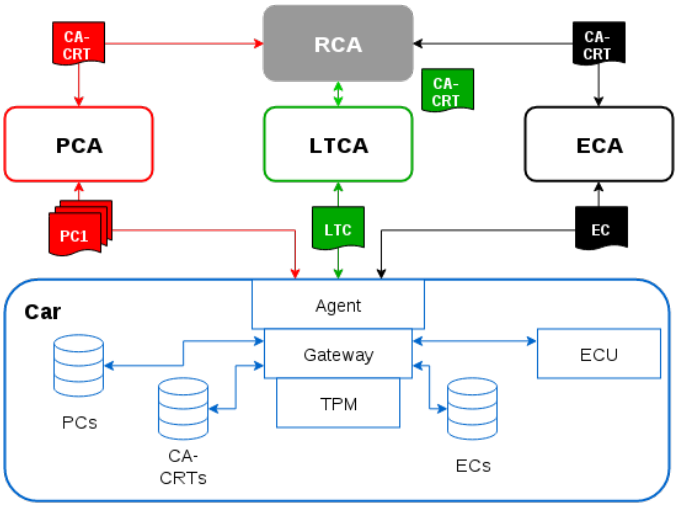


Fig. 6. PKI for architecture

an interface which has a significant limitations for data-rate and frame size. However, due to a fact that German Federal Department for Security in Information Technology already has specified TLS protocol [19] for a very constrained environment – Wireless MBus [20], the TLS-based interface is considered as being a suitable example of well studied security solution which can be used for internal communications; e.g., to exchange symmetric keys and certificates.

Figure 6 depicts an overall PKI architecture for Car2X and on-board communications.

B. Car interaction with Long Term CA

Long Term CA generates its public/private key pair and sends a certification request to the Root CA to get certified. It then has to prove whether it is allowed to sign Long Term certificates (LTC) for cars. LTCs may be created during the manufacturing process of gateways or are introduced into the car at the end of the production line. Hence, either automotive suppliers or manufacturers are considered as possible LTCA operators for cars. But in the former case the supplier must already know the complete vehicle information. The certification process is summarized in Figure 6. The Root CA certifies a CA certificate for LTCA, which certifies an LTC for the car. Because the car needs to know with which CAs it can communicate, CA certificates of trusted CAs are preloaded into

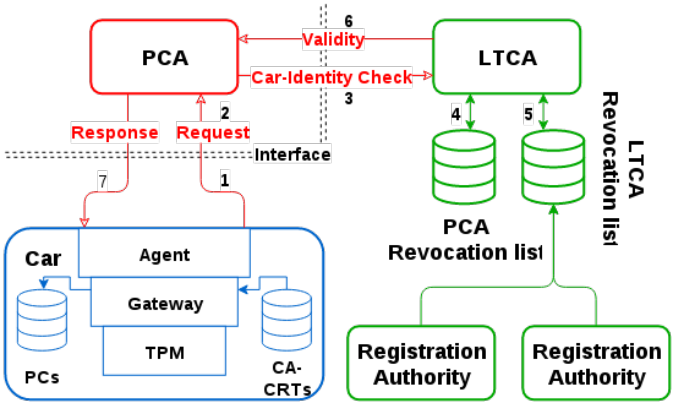


Fig. 7. Pseudonym generation process

an internal certificate store of the car. The Long Term CA is a server component that communicates with a client agent on the car or with other CAs (Figure 7). Hence the LTCA software receives requests, processes them and generates responses. This is illustrated in Figure 7. For the LTC generation, the LTCA receives a request containing: the vehicle information, the public key to be certified, a signature of the request generated with the private key.

LTC generation (initial, before expiring, after revocation): In the LTC generation process, Figure 7, the LTCA receives a certification request. After the verification of the request and its authorization, a certification response containing a signed certificate is stored locally and then returned back to the requester. The detailed generation process can be found in [21] and [22].

C. Car interaction with Pseudonymous CA and Long-term CA

As depicted in Figure 7, the Root CA certifies a CA certificate for the Pseudonym CA, which is responsible for certifying pseudonym certificates for cars. The certification process of pseudonyms is discussed later in this section. The car chooses a PCA from its database and sends the request to it. After getting the PCs, they are stored in the internal certificate store.

As the LTCA, the PCA is a server component, which receives requests from cars and generates responses as illustrated in Figure 7. Pseudonym generation: Because it is recommended that PCA and LTCA are separate entities, the LTCA must also be involved in the pseudonym generation process, Figure 7.

1) The car generates a pseudonym certification request, signs it and sends it to the PCA.
2) The PCA receives the request and checks the region ID:
   a) If the PCA is not the appropriate CA for the request, it is forwarded to the next PCA. Alternative: the PCA returns the address of the appropriate CA to the car. This avoids traffic between PCAs in areas close to frontiers.
   b) If the PCA is the appropriate CA, a forwarding request containing the original one is created, signed and sent to the LTCA.
3) The LTCA receives the forwarding request.
4) The LTCA checks the authenticity of the PCA and the validity of its certificate.
5) The LTCA checks the integrity of the original request and the validity of the associated LTC. The LTCA checks if the LTC has the requested permissions.
6) Finally a response is sent back to the PCA encrypted with its public key.
7) After getting the response of the identity check, if it is valid, then the PCA generates the pseudonym certificates from the received set of public keys. Finally the PCA generates a response message containing the pseudonym certificates and sends it back to the car.
8) The car receives the set of pseudonyms and saves them in its certificates store.

D. Car and ECU interaction. In-car interactions.

There are further certificates needed to secure the in-car-communication between the gateway and the ECUs and between ECUs among themselves. It must be ensured that an attacker cannot connect a fake ECU to the system. This ECU certificate (EC) may contain an identification number for the unit, the public key and the authorization of the ECU [23]. To be able to check this certificate, the gateway holds the certificates of all trusted ECUs in its certificate store. Due to this requirement, the original architecture of C2C-CC needs to be extended.

1) Alternative 1 – external ECU CA (ECA): One possible solution is shown in Figure 8: The PKI described for C2C-CC is expanded by a further ECA. The ECA digitally signs the certificate of the ECU and the gateway can verify it with the certificate of the corresponding ECA from its certificate store.

2) Alternative 2 – In-car-PKI: Another possible solution would be to use the EC from the ECU branch in Figure 8 only once like a quality seal for authorization at the first installation and then to switch to the use of an in-car PKI as shown in Figure 8: After that an in-car-PKI is established in the following way: Prerequisites: 1) The gateway has its firmware installed; 2) The gateway is equipped with means to authenticate service commands from an outside service entity; 3) The new ECUs have their firmware installed and accept connections from the gateway (without authentication); 4) The ECU has a valid certificate for authorization

With a following assumption – the in-car communication is allowed to be unsecured for the duration of the above procedure. The car manufacturer or service facility (service entity) is issuing an authenticated command to the gateway triggering it to accept new ECUs. The gateway is connecting to every new ECU (enumeration technique out of current scope) and is executing the following procedure:
ECU with Trusted Platform Module: Upon the gateways connection request the ECU generates a public/private key pair and forwards the public key to the gateway.
ECU without TPM: The gateway generates a public/private key pair and forwards the private key to the ECU.
In either case: The gateways in-car RCA issues and signs a certificate for the ECU containing its public key and some identifier (e.g. network interface address). The certificate is forwarded to the ECU together with the gateways in-car root and communication certificates.

As was indicated previously in Chapter V-A in order

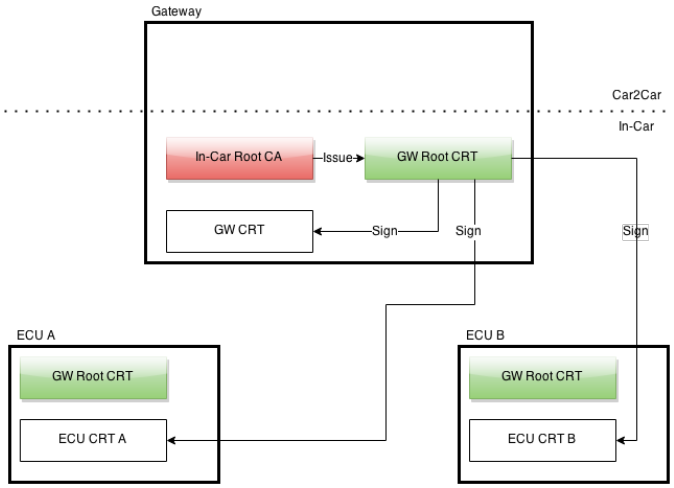


Fig. 8. PKI for in-car systems

to establish CAN bus based secure communication between ECUs or Gateway and ECU, the TLS protocol was taken as a base. However the broadcasting nature of CAN bus interface can't be mapped directly to the TLS protocol, as well as its' stateless delta version - DTLS protocol, due to they use unicast addressing scheme. To overcome this limitation various schemes based on additional transport layer was considered; e.g. widely used in AUTOSAR framework ISO TP standard [24].

VI. Summary and Outlook

We presented a brief and compressed overview of available security related projects for Car2X and on-board communications, as well as an Autosar – basic platform for in-car domain applications. The main contributors for standardization bodies are various pilot projects; e.g., C2C-CC and PRESERVE. However there are plenty of uncovered issues and divergences between different proposals; e.g., DTS application area where on-board ECU needs to be included in a global PKI.

Currently work is ongoing to define a prototype architecture which supports both Car2X and on-board domain, and use standardized approaches like (D)TLS for CAN bus/Ethernet communication within a vehicle.

Acknowledgment

This work was supported in part by the Federal Ministry of Economics and Energy as a cooperative ZIM-KF project under grant number KF2471315LF4. The authors are grateful for this support. They are also grateful for the good cooperation with the project partner Apsec GmbH

References

[1] ETSI TS, "Intelligent transport systems (its); vehicular communication; basic set of applications; part 1: Functional requirements," ETSI TS 102 637-1, European Telecommunications Standards Institute, Tech. Rep., 2010.

[2] ETSI ITSWG, "Intelligent transport systems (ITS); security; threat, vulnerability and risk analysis (TVRA)," ETSI TR 102 893, European Telecommunications Standards Institute, Tech. Rep., 2010.

[3] T. Schütze, "Automotive security: Cryptography for car2x communication," in *Embedded World Conference*. Citeseer, 2011.

[4] C. Miller and C. Valasek, "Adventures in automotive networks and control units," in *DEF CON 21 Hacking Conference. Las Vegas, NV: DEF CON*, 2013.

[5] "IEEE Standard for Wireless Access in Vehicular Environments Security Services for Applications and Management Messages," *IEEE Std 1609.2-2013 (Revision of IEEE Std 1609.2-2006)*, pp. 1–289, April 2013.

[6] A. Festag, "Cooperative intelligent transport systems standards in europe," *Communications Magazine, IEEE*, vol. 52, no. 12, pp. 166–172, 2014.

[7] ETSI TS, "Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management. Technical Specification," ETSI TS 102 940, European Telecommunications Standards Institute, Tech. Rep., 2012.

[8] ——, "Intelligent Transport Systems (ITS); Security; Trust and Privacy Management. Technical Specification," ETSI TS 102 941, European Telecommunications Standards Institute, Tech. Rep., 2012.

[9] F. Armknecht, A. Festag, D. Westhoff, and K. Zeng, "Cross-layer privacy enhancement and non-repudiation in vehicular communication," in *Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference*. VDE, 2007, pp. 1–12.

[10] J. Bishop and J. Nordvik, "Digital tachograph system european root policy."

[11] M. Colak *et al.*, *Cryptographic security mechanisms of the next generation digital tachograph system and future considerations*. Publications Office, 2012.

[12] N. Bißmeyer *et al.*, "Preparing secure vehicle-to-x communication systems," 2014.

[13] T. Leinmüller *et al.*, "Sevecom-secure vehicle communication," in *IST Mobile and Wireless Communication Summit*, no. LCA-POSTER-2008-005, 2006.

[14] B. Weyl *et al.*, "Evita deliverable d3. 2 secure on-board architecture specification," 2010.

[15] O. Henniger *et al.*, "Securing vehicular on-board it systems: The evita project," in *VDI/VW Automotive Security Conference*, 2009.

[16] F. Kargl *et al.*, "Preciosa d7 v2x privacy verifiable architecture," 2010.

[17] "AUTOSAR Release 4.2.2 Layered Software Architecture [Online]," http://www.autosar.org/fileadmin/files/releases/4-2/software-architecture/general/auxiliary/AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf, Accessed: 2015-09-11.

[18] "AUTOSAR Release 4.2.2 Secure Onboard Communication [Online]," http://www.autosar.org/fileadmin/files/releases/4-2/software-architecture/safety-and-security/standard/AUTOSAR_SWS_SecureOnboardCommunication.pdf, Accessed: 2015-09-11.

[19] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246 (Proposed Standard), Internet Engineering Task Force, Aug. 2008, updated by RFCs 5746, 5878, 6176, 7465, 7507, 7568. [Online]. Available: http://www.ietf.org/rfc/rfc5246.txt

[20] OMS Technical Report Security, "Technische Richtlinie BSI TR-03109-1 III, Feinspezifikation Drahtlose LMN-Schnittstelle [Online]," https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03109/TR-03109-1_Anlage_Feinspezifikation_Drahtlose_LMN-Schnittstelle-Teil2.pdf?__blob=publicationFile, Tech. Rep., Accessed: 2015-09-11.

[21] H. Stübing, *Multilayered security and privacy protection in Car-to-X networks: solutions from application down to physical layer*. Springer Science & Business Media, 2013.

[22] A. Khalique, K. Singh, and S. Sood, "Implementation of elliptic curve digital signature algorithm," *International Journal of Computer Applications*, vol. 2, no. 2, pp. 21–27, 2010.

[23] M. Wolf, A. Weimerskirch, and C. Paar, "Secure in-vehicle communication," in *Embedded Security in Cars*. Springer, 2006, pp. 95–109.

[24] "Road vehicles – Diagnostic communication over Controller Area Network (DoCAN) – Part 2: Transport protocol and network layer services," International Organization for Standardization, Geneva, CH, Standard, Mar. 2011.

Testing Embedded TLS Implementations Using Fuzzing Techniques and Differential Testing

Andreas Walz, Axel Sikora
Laboratory for Embedded Systems and Communication Electronics
Offenburg University of Applied Sciences
Offenburg
Email: {andreas.walz, axel.sikora}@hs-offenburg.de

***Abstract*—Security in IT systems, particularly in embedded devices like Cyber Physical Systems (CPSs), has become an important matter of concern as it is the prerequisite for ensuring privacy and safety. Among a multitude of existing security measures, the Transport Layer Security (TLS) protocol family offers mature and standardized means for establishing secure communication channels over insecure transport media. In the context of classical IT infrastructure, its security with regard to protocol and implementation attacks has been subject to extensive research. As TLS protocols find their way into embedded environments, we consider the security and robustness of implementations of these protocols specifically in the light of the peculiarities of embedded systems. We present an approach for systematically checking the security and robustness of such implementations using fuzzing techniques and differential testing. In spite of its origin in testing TLS implementations we expect our approach to likewise be applicable to implementations of other cryptographic protocols with moderate efforts.**

I. Introduction

Given today's pervasion of Information Technology (IT) and in particular embedded Cyber Physical Systems (CPSs), any flaw in or malfunction of such systems can cause economic loss, impair privacy, or can even be dangerous to life [1]. To ensure safety and privacy, many security measures have been proposed, standardized, and implemented [2]. Among them is the family of Transport Layer Security (TLS) protocols[1] [3], [4], [6] which offer mature and standardized means for establishing secure end-to-end communication channels over insecure transport media. TLS supports encrypted and integrity-checked data transfer as well as peer authentication using a Public Key Infrastructure (PKI) or pre-shared keys.

The security of TLS protocols [10], [15] as well as their implementations has been the subject of various research activities [7], [8], [9], while implementations have mostly been studied in the context of classical IT infrastructure. However, as the use of TLS protocols in embedded environments receives increasing attention it is vital to consider the security of TLS implementations specifically in the light of the peculiarities of embedded systems. From an application's perspective these are mainly constrained resources (e.g. computing power, memory, bandwidth, etc.) and physical exposure to potential attackers [23], [24]. From a tester's perspective it is primarily the lack of efficient and convenient monitoring and control mechanisms that are typically required when applying common software testing techniques. To address this, we are developing and implementing an approach to systematically test the security and robustness of embedded TLS implementations using fuzzing techniques and differential testing. It treats implementations as black boxes and does not require any memory or execution monitoring. Our approach somewhat resembles the concept of Frankencerts [7], though leveraging its ideas to be applicable to entire TLS protocol implementations rather than their certificate validation logic alone. While seeking to automate the process it is expected to feature decent coverage. We consider our approach supplementing other, more formal or semi-formal approaches.

The paper is structured as follows: after this introduction Section II will briefly present work related to ours. The TLS protocol family will be introduced in Section III. Sections IV and V present our test approach and the framework implementing it, respectively. A summary and an outlook is given in Section VI.

II. Related Work

Several generic approaches to test implementations of communication protocols have been presented in the past [16], [17], [18], [19], [20]. These approaches use random or semi-random mutations to generate potentially invalid input and various different techniques to identify invalid responses or behaviour. Among these are response validation by means of formal specifications or monitoring the system's memory and execution to detect abnormal program behaviour.

SECFUZZ is a tool that specifically addresses the complications induced by the use of cryptography when testing implementations of security protocols using fuzzing [21]. As others do, it uses memory monitoring to detect abnormal program behaviour.

Frankencerts is a tool and an approach to specifically test the validation logic of X.509 certificates embedded in TLS implementations [7]. It combines fuzzing techniques for generating randomized certificates and differential testing used as an oracle for discovering incorrect certificate validation in TLS implementations. Frankencerts only considers certificates as mutable input but does not alter TLS messages themselves.

Other approaches to identify flaws in implementations of the TLS protocol state machines have been proposed and implemented successfully in the past [8], [9]. With these approaches, manipulated TLS messages are used to probe TLS implementations for the possibility to trigger invalid transitions

[1] In this paper, we use the plural form *TLS protocols* to refer to the diversity of different versions and variants of TLS (including DTLS).

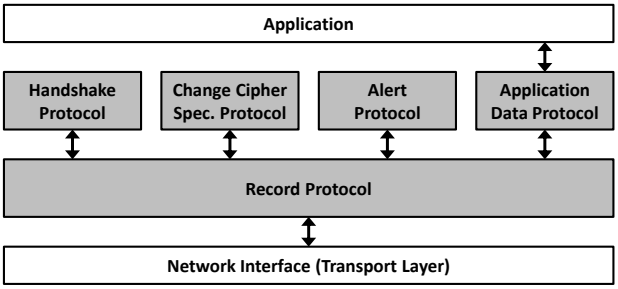


Fig. 1. Illustration of the (D)TLS protocol stack with its fundamental Record protocol as well as its four higher-layer sub-protocols for connection establishment (Handshake and Change Cipher Spec protocol), error signaling (Alert protocol) and application data transmission (Application Data protocol).

in the state machines implementing the TLS protocol. Invalid transitions are identified either by explicitly crafting offensive input or by manual inspection of derived state machine models.

Formal verifications of TLS implementations are to the best of our knowledge only performed for implementations specifically designed for this purpose [26], [27].

III. The TLS Protocol Family

The TLS protocol family provides a flexible framework for cryptographically securing connections from end to end over communication channels assumed to be under an active attacker's control. TLS is the successor of the Secure Sockets Layer (SSL) protocol [5] introduced in the 1990s [6] and has become the de-facto standard for secure internet communications. Since its outset, TLS evolved from version 1.0 to its currently latest version 1.2 [3] incorporating important security amendments. As TLS requires reliable transport (e.g. via TCP) Datagram TLS (DTLS) [4] has been designed to accommodate the specifics of unreliable transport media (e.g. via UDP). The structure and concept of DTLS closely resembles the ones of TLS but it introduces several important changes in the protocol's communication format and flow.

A. Structure of TLS

TLS (and DTLS) is connection-oriented and follows a client/server model with well-defined roles. It is composed of five sub-protocols as indicated in Figure 1. The fundamental protocol is the Record protocol. It is fragmenting data from higher layers and handling data encryption as well as generation and verification of Message Authentication Codes (MACs) to ensure data integrity. Its protocol data unit (PDU) is called a TLS record. Stacked on top of the Record protocol are the Handshake protocol, the Change Cipher Specification protocol, the Alert protocol, and the Application Data protocol. The first two are responsible for establishing and negotiating TLS connection parameters and keys. The Alert protocol delivers error messages and the Application Data protocol is used to transfer raw application-layer data.

TLS does not use fixed cryptographic algorithms but introduces the concept of cipher suites to offer a multitude of possible combinations of cryptographic algorithms and schemes. Each cipher suite selects a mechanism for key exchange and authentication, an algorithm for message integrity checks, and

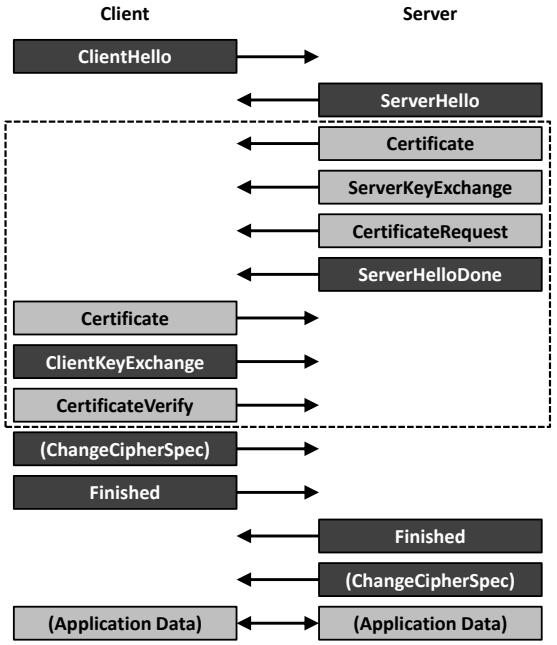


Fig. 2. Illustration of the sequence of messages exchanged between a TLS client and a TLS server during the handshake. Messages with a solid dark background are mandatory for a full handshake whereas messages with a light background are optional or context-dependent. Messages correspond to the Handshake protocol unless they are shown in parentheses. Messages within the dashed box are omitted for an abbreviated handshake (session resumption).

a symmetric algorithm for bulk data encryption. Furthermore, an extension mechanism allows clients and servers to negotiate a multitude of optional features that are not part of the core protocol standard, giving TLS an outstanding degree of flexibility.

Any TLS connection is commenced by a handshake composed of a sequence of binary-encoded messages exchanged between the client and the server. It is triggered by the client and initiated by the exchange of two *Hello* messages facilitating – amongst other things – the negotiation of protocol parameters. Authentication of the communication partners and the establishment of a shared secret between them are achieved by the exchange of dedicated *Certificate* and *KeyExchange* messages. Finally, *Finished* messages complete the handshake and confirm its authenticity in retrospect.

A full TLS handshake involves asymmetric cryptographic operations for authentication and key establishment. Depending on the configuration and the type of Public-Key Infrastructure (PKI) in place different cryptographic algorithms such as RSA, Diffie-Hellman (DH), DH with ephemeral keys (DHE), or the Digital Signature Algorithm (DSA) may be used. For DH(E) and DSA, also variants based on Elliptic Curve Cryptography (ECDH, ECDHE, ECDSA) are specified. Session resumption omits authentication and key establishment by reusing a previously negotiated shared secret.

Figure 2 illustrates the sequence of messages exchanged between a TLS client and server during the handshake. Each message is a complex structure of binary-encoded integer and enumeration fields as well as nested sub-structures. Additionally, cryptographic elements (signatures, random values, etc.)

might be present.

B. Security of TLS and its Implementations

The formal and abstract security of TLS protocols has been studied extensively in the past [11], [12], [13], [14], [15], [10] and many attacks and weaknesses in by now outdated protocol versions and algorithms have been identified. However, to the best of our knowledge TLS in its latest version can be considered secure – assuming it is implemented correctly and used with appropriate parameters.

Not least is the possibility of fixing security issues in TLS attributable to its flexibility. However, this flexibility comes at the expense of simplicity and implementations of the protocol rapidly become quite complex as they seek to cover the multitude of existing protocol variants, cipher suites, and extensions. This fact enhances the potential for flaws in implementations of TLS that impair the protocol's abstract security claims. Unsurprisingly, several studies have uncovered serious security issues in popular TLS implementations [7], [8], [9]. Virtually all of them were related to an insufficient validation of input received over the network. Therefore, it is crucial to confirm an implementation's resilience against input maliciously crafted to exploit certain implementation flaws.

IV. Test Approach

Our approach strives to identify flaws in TLS implementations deployed in embedded environments while it is minimizing the need for manual intervention during the test process. It focuses on automatically discovering unknown issues rather than checking against common or known issues retrieved from a suitable database. It treats TLS implementations under test as black boxes and avoids the need for sophisticated execution and memory monitoring. Furthermore, it does not impose any requirements on the application operating on top of the TLS stack to be tested, though certain application characteristics might enable supplemental test features.

Our approach combines two techniques that by themselves have already been used before (see also Section II). Fuzzing is used to generate potentially invalid input and differential testing is used as an oracle to detect improper system reaction. In this respect it can be considered an extension of Frankencerts [7], leveraging its concept to be applicable to entire TLS protocol implementations rather than their certificate validation logic alone.

A. Input Generation

Fuzzing involves monitoring a system for malfunction while being passed randomly or semi-randomly generated input that is likely to violate the system's input specification [22]. Fuzzing approaches come in various different flavours, while we make use of mutative fuzzing drawing message templates from pre-recorded and sufficiently diverse TLS traffic. Based on these message sequences, selection, mutation, manipulation, and adaption operations are applied. This is done such that it implicitly emulates the behaviour of either a TLS client or server without explicitly implementing the TLS protocol.

For every message to be sent to the TLS implementations under test a new message is selected from the database. Messages are selected preferentially from the same sequence in the same order they have been recorded. However, with configurable probabilities messages are selected in a different order, from a different but compatible sequence, or even from completely incompatible sequences. Compatibility between message sequences is evaluated using a dedicated metric based on heuristics accounting for protocol parameters, algorithms, and roles that have been used in the recorded protocol run.

Mutations applied to selected messages involve reordering similar consecutive objects within a message (e.g. in an array of fields), removing objects from the message, multiplicating existing objects, or inserting objects taken from another message. Manipulations involve selectively changing fields in the protocol message by flipping bits, changing the value of fields, or setting fields to invalid values. All these operations are performed on multiple levels of nested message fields and in multiple steps with configurable probabilities. As some fields in protocol messages might be interrelated by consistency requirements (e.g. fields that encode the length of other parts of the message) these fields are updated after the mutation and manipulation processes to restore consistency. Exceptions to this are manipulations explicitly targeting such consistency relations.

As the TLS Record protocol allows fragmentation and coalescence of messages from higher-layer TLS sub-protocols the corresponding degrees of freedom are likewise subject to manipulation and variation.

B. Response Analysis

Differential testing makes use of one or more additional systems that are supposed to behave identically but due to structural or implementation differences bear the potential to exhibit discrepant behaviour upon receiving equivalent input. In such a setup, discrepant behaviour either points to an imprecise specification the systems try to follow or – as is more interesting for our purposes – indicates faulty behaviour of at least one system involved. Differential testing makes use of this fact to obtain an oracle for invalid system behaviour without requiring a specification of the expected behaviour. However, it relies on the assumption that there is no flaw that similarly affects all systems involved.

In our approach, the TLS implementation to be tested is accompanied by additional, structurally different TLS implementations of different origins. The evaluation of the consistency of the responses across involved implementations is done by means of a metric similar to the one which is used to evaluate the compatibility of different message sequences from the database. That is, the type of response sent by TLS implementations (e.g. handshake or alert message) as well as the messages' content are compared and the compatibility is evaluated.

C. Destructive State Probing

Among others, an interesting question when testing TLS implementations is whether an attacker is able to fraudulently trick a TLS implementation into accepting and forwarding application data without being in possession of valid permission and/or credentials. Therefore, in any state of the test process – and in particular during an ongoing handshake – application

data is sent to involved TLS implementations with configurable probability for the mere purpose of probing its acceptance. Such premature application data injection would trigger a valid TLS implementation to close the current connection with a fatal alert[2]. We refer to this type of probing as *destructive state probing* as it destroys the current connection state.

D. Dealing with Cryptography

In the context of TLS both the generation of input as well as the use of differential testing are complicated by certain protocol features and TLS' inherent utilisation of cryptography. For each TLS connection session keys inevitably are different unless both client and server use fixed data in lieu of randomness. As only one side is under the tester's control, cryptographic fields have to be recalculated before data is passed to the TLS implementations. Similarly, variations in responses induced by differing cryptographic keys have to be harmonized before data can be compared across implementations.

E. Issue Discovery Capabilities

Our approach is expected to allow to identify a wide range of security-relevant issues in TLS implementations that are somehow related to an improper parsing and/or validation of input received over the network. However, since we refrain from using invasive system monitoring we also expect our approach to miss important issues as they won't express themselves in discrepant system responses, e.g. out-of-bounds read operations in memory.

V. Test Framework

Currently, we are developing a test framework that is going to implement the test approach presented in the previous section. It consists of several components illustrated in Figure 3.

A. Sending Messages

A database (*Message sequence database*) holds the prerecorded TLS message sequences and makes them available to the component (*Message selection / test control*) which is responsible for selecting messages for further processing. Selected messages are passed to the component (*Message mutation/manipulation*) that is applying mutation and manipulation operations before passing them to the component (*Crypto adaptor*) which is transparently injecting and/or harmonizing cryptographic dependencies. The crypto adaptor is passing messages to the component (*Encoder / decoder*) which is handling message encoding and decoding. This also includes handling message fragmentation and coalescence. As TLS protocols are multi-layered, messages might need to circulate multiple times between the latter two components before being dispatched to the TLS implementations under test.

[2]DTLS, unfortunately, does not behave in this way necessarily. Therefore, this kind of probing is only applicable to DTLS if the corresponding application allows to feedback received application data to the test system.

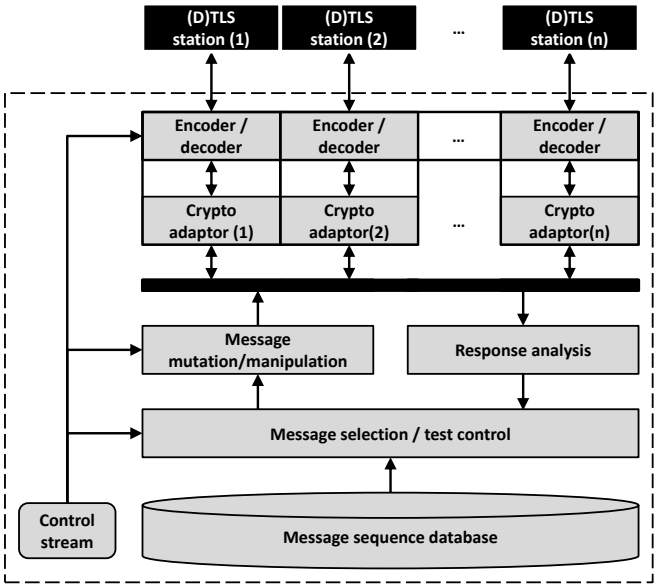


Fig. 3. Architecture of the framework for testing implementations of a TLS protocol. The framework's boundary is indicated as a dashed box.

B. Receiving Messages

Messages sent by the implementations under test are processed by the encoder / decoder and crypto adaptor components before being passed to the component (*Response analysis*) which is evaluating and comparing the systems' responses. The results of this evaluation are fed back to the component (Message selection / test control) controlling the test process.

C. General Message Handling

Our strategy for generating input (message mutation, manipulation, adaption) and analysing responses requires a machine-readable description of the format and encoding of TLS messages. Unfortunately, the on-the-wire representation of messages tends to impede the implementation of such operations in a way that is both simple and flexible at the same time. This is mostly due to the necessity of implementing complex and protocol-specific code for efficiently translating between the raw binary representation and the structured high-level representation of messages. Furthermore, handling data on bit-level in high-level programming languages is a tedious and error-prone task. Therefore, as an integral component of our test framework we are developing a *C++*-based framework that starting from the description language defined for specifying TLS [3] automatically generates source code for translating between packet representations.

In supporting the encoding, decoding, generation, and dissection of TLS messages, our message handling tool is similar to *FlexTLS* [25], a tool for rapidly prototyping and testing implementations of TLS protocols. However, our tool aims for providing protocol-agnostic services to easy our test approach's adaptability to other security protocols.

D. Test Control

As indicated in Section IV, the test process is semi-random and depends on several configurable probabilities. Each decision within the process that is based on probabilities is come to by evaluating a continuous stream of opaque control data. Blocks of data are read from this stream and interpreted as encoding integer values. Probabilities are implemented by accordingly mapping decisions to corresponding ranges of these integer values. The opaque stream of control data is provided by a component (*Control stream*) which is delivering either random or semi-random data or – in order to enable reproducibility and determinism – taking data from a deterministic input, e.g. a stored file.

VI. Summary and Outlook

We presented an approach for discovering flaws in implementations of TLS protocols in embedded environments. Pre-recorded TLS traffic is semi-randomly – though purposefully – mutated to provoke faulty behaviour in these implementations. This input generation strategy does not explicitly but most likely produce invalid input. Instead of examining responses by means of verified references or monitoring implementations for abnormal execution states we employ differential testing to detect suspicious behaviour. We expect our approach to be capable of detecting a broad range of input parsing and/or validation issues in TLS implementations as they express themselves in suspicious system responses.

Currently, we are implementing the test framework and refining our test approach including evaluation metrics and input generation strategies. A theoretical study of the approach's capabilities as well as their dependence on test parameters is ongoing. Furthermore, we are studying the adaptability of our approach to other cryptographic protocols.

We are going to apply our approach to a list of popular as well as a proprietary TLS implementation that are individually running an small embedded platforms to experimentally assess its effectiveness.

Acknowledgment

This work is partially funded by the Ministry of Economic Affairs and Energy in the framework of the ZIM program within the project "Sicherheitskomponenten für Industrie 4.0 Lösungen Angriffsvektoren, Open Source-basierte Sicherheitsmodule und industrietaugliche Lifecycle Komponenten Sind4.0" (KF2471320KM4). The authors are grateful for this support.

References

[1] H. Gravel and S. Graf, *Formal Methods for Safe and Secure Computer Systems*, Bundesamt für Sicherheit in der Informationstechnik, 2013.

[2] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 5th Edition, Pearson Education, 2010.

[3] T. Dierks and E. Rescorla, *RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2*, Internet Engineering Task Force, 2008.

[4] E. Rescorla and N. Modadugu, *RFC 6347: Datagram Transport Layer Security Version 1.2*, Internet Engineering Task Force, 2012.

[5] A. Freier, P. Karlton, and P. Kocher, *The Secure Sockets Layer (SSL) Protocol Version 3.0*, Internet Engineering Task Force, 2011

[6] R. Oppliger, *SSL and TLS: Theory and Practice*, Artech House Inc., 2009.

[7] C. Brubaker et al, *Using Frankencerts for Automated Adversarial Testing of Certificate Validation in SSL/TLS Implementations*, IEEE Symposium on Security and Privacy, 2014.

[8] B. Beurdouche, K. Bhargavan, A. Delignat-Lavaud, C. Fournet, M. Kohlweiss, A. Pironti, P.-Y. Strub, and J. K. Zinzindohoue, *A Messy State of the Union: Taming the Composite State Machines of TLS*, IEEE Symposium on Security and Privacy, 2015.

[9] J. de Ruiter and E. Poll, *Protocol State Fuzzing of TLS Implementations*, USENIX Security Symposium, 2015.

[10] C. Meyer and J. Schwenk, *Lessons Learned From Previous SSL/TLS Attacks – A Brief Chronology Of Attacks And Weaknesses*, Cryptology ePrint Archive, 2013.

[11] L. C. Paulson, *Inductive Analysis of the Internet Protocol TLS*, ACM Transactions on Information and System Security, 1997.

[12] G. Díaz, F. Cuartero, V. Valero, and F. Pelayo, *Automatic Verification of the TLS Handshake Protocol*, ACM Symposium on Applied Computing, 2004.

[13] K. Ogata and K. Futatsugi, *Equational Approach to Formal Analysis of TLS*, International Conference on Distributed Computing Systems, 2005.

[14] S. Gajek, M. Manulis, O. Pereira, A.-R. Sadeghi, and J. Schwenk, *Universally Composable Security Analysis of TLS – Secure Sessions with Handshake and Record Layer Protocols*, International Conference on Provable Security, 2008.

[15] H. Krawczyk, K. G. Paterson, and H. Wee, *On the Security of the TLS Protocol: A Systematic Analysis*, Advances in Cryptology – CRYPTO 2013, 2013.

[16] G. Wimmel and J. Juerjens, *Specification-Based Test Generation for Security-Critical Systems Using Mutations*, International Conference on Formal Engineering Methods, 2002.

[17] G. Banks, M. Cova, V. Felmetsger, K. Almeroth, R. Kemmerer, and G. Vigna, *SNOOZE: Toward a Stateful NetwOrk prOtocol fuzZEr*, Information Security, 2006.

[18] D. Lee and G. Shu, *Testing Security Properties of Protocol Implementations – a Machine Learning Based Approach*, International Conference on Distributed Computing Systems, 2007.

[19] G. Shu, Y. Hsu, and D. Lee, *Detecting Communication Protocol Security Flaws by Formal Fuzz Testing and Machine Learning*, Formal Techniques for Networked and Distributed Systems – FORTE 2008, 2008.

[20] Y. Hsu, G. Shu, and D. Lee, *A model-based approach to security flaw detection of network protocol implementations*, International Conference on Network Protocols, 2008.

[21] P. Tsankov, M. T. Dashti, and D. Basin, *SECFUZZ: Fuzz-testing security protocols*, International Workshop on Automation of Software Test 2012, 2012.

[22] R. McNally, K. Yiu, D. Grove, D. Gerhardy, *Fuzzing: The State of the Art*, Technical Report, Australien Government, Department of Defence, 2012.

[23] D. N. Serpanos and A. G. Voyiatzis, *Security Challenges in Embedded Systems*, Transactions on Embedded Computing Systems,c 2013.

[24] S. Jaeckel, N. Braun, and A. Sikora, *Design strategies for secure embedded networking*, Long-Term and Dynamical Aspects of Information Security, 2007.

[25] B. Beurdouche, A. Delignat-Lavaud, N. Kobeissi, A. Pironti, and K. Bhargavan, *FLEXTLS: A Tool for Testing TLS Implementations*, USENIX Workshop on Offensive Technologies, 2015.

[26] K. Bhargavan, C. Fournet, M. Kohlweiss, A. Pironti, and P.-Y. Strub, *Implementing TLS with Verified Cryptographic Security*, IEEE Symposium on Security and Privacy, 2013.

[27] D. Kaloper-Mersinjak, H. Mehnert, A. Madhavapeddy, and P. Sewell, *Not-quite-so-broken TLS: lessons in re-engineering a security protocol specification and implementation*, USENIX Security Symposium, 2015.

Towards Privacy for Ambient Assisted Living in a Hybrid Cloud Environment

Hendrik Kuijs, Christoph Reich, Martin Knahl
Institute for Cloud Computing and IT Security
Furtwangen University
Furtwangen, Germany
Email: {Hendrik.Kuijs, Christoph.Reich, Martin.Knahl}@hs-furtwangen.de
Nathan Clarke
Centre for Security, Communications and Network Research
Plymouth University
Plymouth, United Kingdom
Email: {Nathan.Clarke}@hs-furtwangen.de

***Abstract*—Ambient Assisted Living (AAL) introduces technology in the living environment of elderly people to support them in their daily life and to prevent emergency situations. This is done by the use of middleware-systems that have to be installed in a user's home. Recent projects consider cloud platforms as more cost effective and flexible for delivering services for AAL. In this paper we present the status of research for a privacy preserving way of delivering services through a Platform as a Service in a hybrid cloud environment. Commonly used concepts are transferred to our platform approach and particular differences are presented.**

***Index Terms*—PaaS, AAL, Hybrid Cloud, OSGi, Privacy**

I. Introduction

Due to the increasing average life expectancy and the decreasing birth-rate the proportion of the young working population in developed countries world-wide is shrinking continuously [1]. Extended families that are able to care for their elderly relatives are slowly disappearing and families are getting smaller in general. Programs for new nursing or day-care facilities are introduced by politics, but there is still a lack of trained personnel and existing facilities. To keep the needed time-span for professional care facilities minimal, with AAL new technological concepts are introduced in the known living environment of the elderly people and people in need of help. As surveys show, elderly people want to stay at home as long as possible [2] and therefore support this trend.

AAL platforms are often based on concepts of smart-home environments that combine input-devices like sensors, user-interfaces or cameras and output-devices like alarms, emergency call functionality, databases or graphical user interfaces. This is provided by a configurable middleware and compute-power that has to be installed in the user's living environment. If new demanding services are introduced into the smart home environment, the existing computing equipment has to be renewed or extended as well.

Our research focuses on delivering cloud based services for AAL. Cloud computing enables service providers (e.g., care givers or day care facilities) to deliver services without the need for investing in expensive technical equipment in advance. By delivering services through the cloud, the high start-up costs can be reduced significantly and it will be feasible for service providers and users to try out new or innovative services without the need of a high investment.

This flexibility can be provided by a customizable Platform as a Service (PaaS) that is run by the service provider. The PaaS is considered to run in a private cloud, as adaptation of the system to the user's need is heavily based on personal user data. The downside of this setting is that it will not scale beyond the boundaries of the physical hardware that is used for running the private cloud.

For this the context-aware PaaS is run in a private cloud, that can be extended with services in the public cloud for better flexibility, scalability and maintainability according to the definition of a hybrid cloud [3]. This hybrid cloud approach poses new challenges for personalisation of services, as personal information is often not allowed to be passed to third parties. Privacy and security constraints have to be considered and methods for providing adaptation of services while preserving privacy have to be introduced in the PaaS for AAL.

Section II gives a brief overview of existing platforms and first approaches to introducing cloud service for AAL. The platform speciAAL is introduced in section III and followed by the main motivations for running the platform in a hybrid cloud environment in section IV. First approaches to privacy when delivering services outside of the Private Cloud are presented in section V. Section VI gives a conclusion and a further outlook of upcoming research topics.

II. Related Work

Existing platforms in the field of AAL are mainly focused on delivering customizable middleware for smart home environments and the whole computing power has to be installed in the environment itself. Examples are SOPRANO [4], AMIGO [5], ProSyst [6] and universAAL [7] that are providing middleware and techniques to gather user data, do reasoning based

on ontologies and recognize events or incidents that lead to corresponding system reactions.

Kim et al. [8] present a platform approach to share health data in the cloud in a secure way. The patient-centric solution provides strong security and privacy characteristics and is entirely governed by the patient. It allows sharing of health data between hospitals, trained care-personnel or relatives to indicate changes in the health conditions amongst different support groups of the user.

Ekonomou et al. [9] developed an extensible OSGi-based architecture for highly heterogeneous smart home systems. This architecture is focused on the integration of new devices by using a cloud-based service for discovering drivers in a manual, semi-automatic and automatic way.

The project Cloud-oriented Context-aware Middleware in Ambient Assisted Living (CoCaMAAL) [10] moves the context generation and classification of incidents in the cloud. Data of installed sensors and devices in the smart living environment is collected by a *Data Collector* on-site and transferred to a context aggregator in the cloud, that sends back appropriate actions.

III. SpeciAAL

The security and privacy enhanced cloud infrastructure for AAL (speciAAL) focuses on delivering personalised and adapted services for information, communication and learning. It is based on the project Person Centered Environment for Information Communication and Learning (PCEICL) [11] which is developed in the Collaborative Centre for Applied Research on Ambient Assisted Living (ZAFH-AAL) [12].

The Plattform as a Service (PaaS) is based on OSGi. OSGi supports installing, starting and stopping software bundles during runtime. Dependencies are managed and resolved by BND-tools, so that dependent services are installed and started prior to starting a new service. This provides the flexibility that is needed for a platform in terms of extensibility and updatability during run-time without the need to restart the whole environment.

The communication inside the platform is performed by software agents with role-based access and communication rights to secure the data-flows inside the platform. These agents communicate based on the Agent Communication Language (ACL) [13] that is backed by an ontology [14] to persist the concepts of passed data or attributes throughout the whole platform.

Information about the user and his environment is stored in an ontology database. This data can be evaluated by software agents for the purpose of adapting a certain behaviour of a service to the supposed need of a user at the current time.

The stored information is gathered by configuration (e.g., during initial installation and configuration of the platform for the user), through connection with existing sensors in a smart-home environment or by data, that is coming from other services inside the PaaS, e.g., a calendar service, a contacts service or a weather service. Ontop of this there is still ongoing research for profile building with machine-learning techniques by analysing user behaviour.

New services can be retrieved and installed through a bundle repository based on OBR [15]. This bundle repository is extended with useful information about the proposed service (e.g., name, description, images or screen shots) and usage information.

The whole management of the platform, the management of services, the communication between platforms and with the underlying Infrastructure as a Service (IaaS) is centralized in the AAL PaaS Manager.

SpeciAAL continues this research but focuses on security and privacy aspects, when running the PaaS in a hybrid cloud environment.

IV. From local installation to the cloud

As mentioned in section II the majority of platforms for AAL rely on installation in a users home environment. This requires that hardware and software has to be setup for each subscriber of the platform. This is likely associated with high costs prior to convincing a user of the benefit of the platform or the usefulness of a specific functionality for the user. By delivering service through a cloud approach, it becomes feasible to both, the service provider and the user, to test services while keeping the initial installation and configuration costs minimal.

It is assumed, that the service provider for such scenarios would be directly connected with a health care provider (e.g., the german red cross [16], ASB [17] or CARITAS [18]) or even be the health care provider itself. As the health care providers are already working with possible future users (e.g., meals on wheels, nursing, day care and emergency services) they could introduce the AAL platform to the users as well.

The cloud approach would be a Private Cloud hosted by the service provider. This enables scalability and extensibility for the environment as new nodes can be started when needed and the platform is able to grow as new users and new services are added. On the other hand this simplifies the maintenance of the platform, as updates can be applied centrally and don't have to be rolled out at local installations in the users' homes. The privacy aspect in this setting is controlled by a security and privacy layer based on ACL as mentioned in section III. The storage of data and data-flows within the Private Cloud are considered to be within the legal boundaries and are not passed to third parties.

To further reduce costs or to dyncamically adjust to changing resource requirements, it may make sense to go one step further and shift services with higher resource needs to a Public Cloud (e.g., of a third party cloud provider). Due to service level agreements and legal restrictions this means, that a different consideration and decision has to be made: What information is needed by the service for delivering a personalised or adapted service and what information is allowed to be passed to services that are not run in the closed ecosystem of the Private Cloud at the service provider. This contradiction of personalisation and privacy concerns must be

handled by special methods and tools inside the platform as well as before provisioning a service through a service or bundle repository.

V. Possible solutions for privacy in hybrid cloud environments

The directive 95/46/EC of the European Parliament [19] defines personal data as follows:

> Personal data shall mean any information relating to an identified or identifiable natural person ('data subject'); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity.

The 'Article 29 Data Protection Working Party of the EC' [20] further explains in Opinion 4/2007 [21] that this definition covers any information that relates to an identifiable, living individual. The differentiation directly or indirectly identifiable is best described by examples: A person's full name and address is an obvious identifier (direct identifier). But a person can also be identifiable from other information, e.g. hair-colour, shoe-size, usual activities and combinations thereof. If there is too much unspecific information the information can be combined to meaningful information and an individual can be identified. This definition is also technology neutral - it does not matter how the personal data is stored.

For cases where the EC directive is not applicable, national legislation is still applicable and in some cases more specific and restrictive. To keep things simple in the conceptual phase we are focusing on EC jurisdiction.

A. Preconditions for the platform

To determine what data is considered personal or private information of the user, the stored data has to be categorised when building up the ontology. The challenge for this categorisation is to be in accordance to legal requirements. This will make it necessary to not only define what information may or may not be released to third parties (direct identification), but also what information may not be released together with what other information (indirect identification, as mentioned before). These policies are applicable for data transfer to the public cloud, but also for data transfer to external services.

Figure 1 gives an overview of an installation process for a new service in the Public Cloud. It is assumed that the user has triggered the installation process and by some requirements (e.g., shortage of resources or cost efficiency) the new service has to be started in the Public Cloud. The Install Helper then requests the needed data for the initial configuration and adaptation of the service. The Privacy Module checks the AAL Privacy Policy whether there are policy breaches in the requested information. In this scenario no breaches exist and therefore the data may be passed to the Install Helper which then passes it over to the Install Helper in the Public Cloud to install the New Service.

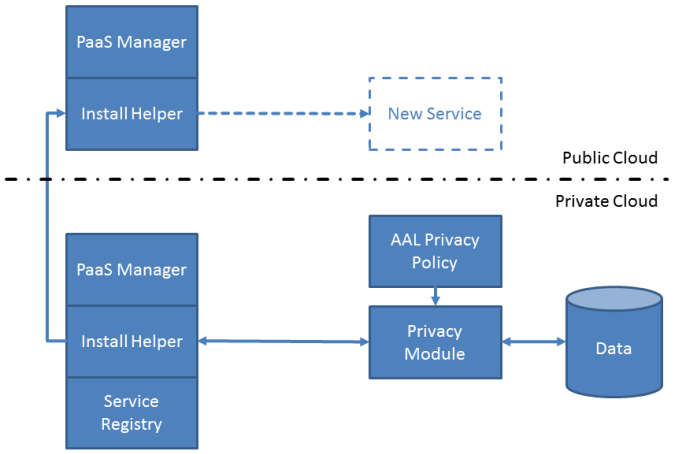


Fig. 1. AAL Privacy Policy and Privacy Module

If the policy check fails, the Install Helper has to request new resources from the PaaS Manager to be able to install the service in the Private Cloud. One exception to this policy driven decision making could be, that the user actively enforces a service to get data beyond this policies as shown in section V-D.

B. Preconditions for services

The simplest service doesn't need any adaptation for the user, because it just delivers information (e.g., a web-server, that has no access tracking in place and just delivers multimedia-content).

As soon as adaptation is necessary, the data that is needed to adapt a new service for a user has to be documented. The suitable places are in the service description and in the manifest-file of each involved OSGi bundle to address the modular characteristics of services based on OSGi. This information is stored in a machine-readable syntax and can be accessed by the Install Helper and Privacy Module and used to determine whether the service is considered to be installed in the private cloud or may be installed in the public cloud.

As an analogy, the federated identity solution and single sign-on service Shibboleth [22] handles these so called attributes that are passed to a third-party service provider by agreements between the identity provider and the service provider. These agreements are added to the attribute-filter list (in our approach the policy), a XML-based description file and is the base information for the identity provider service to pass attributes to the service after authentication.

Private data that is passed to a service has to be kept inside the service and may not be passed to third party application. This has to be guaranteed for the services in a trustworthy environment. In official smart phone ecosystems like the Google Play Store or Apple Appstore this is not only required for developers of new applications but also tested prior for adding the service in the application store. These tests are done in a hybrid manner: By automated testing of the application and by code reviews of the applications. For

a trustworthy platform for ambient assisted living, these tests are required as well.

C. Reevaluating data access

Besides installation and initial configuration, user data is also requested for adapting and reconfiguring a service during run-time. This reconfiguration is triggered by changes in user data that are reasoned by agents and combined in events. An example would be a fitness service with video-trainings that is reconfigured to exclude trainings for feet after the user has sprained his ankle. The service is registered in the service registry (as shown in figure 1) together with the documentation of the needed data for configuration. With this information the Privacy Module can reevaluate if the policies are still met and the updated information is allowed to be passed in the Public Cloud.

This reevaluation is needed because a change in user data can mean that an information that was not defined during installation of a service now has some value and is considered a breach of privacy. In the fitness example above, the service was used with no restriction before the sprained ankle and after the information of the sprained ankle has to be passed for adaptation. If the sprained ankle is defined as personal information (see [21]) the reconfiguration requires the service to be shifted to the Private Cloud.

D. Transparency

One feasible solution for compliance when transferring user data to third parties is transparency. As a legal requirement, a user has to be informed about user information that is passed to third parties. This can be used to 'override' the policies that are evaluated automatically in the Privacy Module.

In Android the user is informed when installing a new application about what sensitive information will be used by the application (e.g. contact details or the whole address book) and the user has to decide if the application is allowed to access the data or not. Sometimes this decision results in not being able to use the application at all. This consequence is not the desired behaviour of services for AAL.

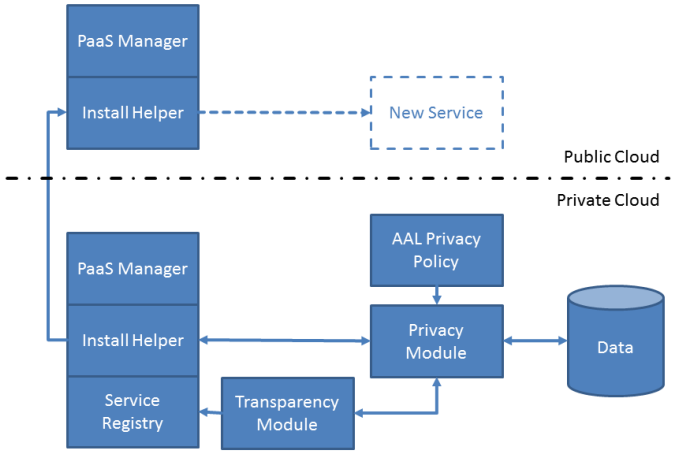


Fig. 2. AAL Privacy Policy and Privacy Module

Figure 2 shows the same scenario as Figure 1 except this time there is a privacy policy violation. A Transparency Module is triggered about the violation and the user is informed about the personal information that will be passed to the service. In this scenario the user complies with the data transfer, the decision is saved in the Service Registry and the New Service is installed and started in the Public Cloud. If the user disagrees, the service cannot be started in the Public Cloud.

The difficulty of transparency in our case is the way in which the information is displayed to the user and the motivation for a user to allow the transfer of data to a third party: If information is to detailed for the user (especially in the field of AAL) the user gets unsettled, confused or annoyed. Especially when a service is expected to be adapted continuously, the user is required to examine each changes or tick the infamous 'Always' box, that is also often used in personal firewall installations when a service changes its connections. The second question that comes in mind is: Why would a user do it in the first place? What is the benefit for the user for having the service running in the Public Cloud (or by lack of concept 'somewhere else')? The considerations of the service provider (reduced costs or extending resources) can't be transferred to the user.

E. Anonymisation or Pseudonymisation

As noted by the Article 29 Data Protection Working Party [21] anonymisation and pseudonymisation can be used to avoid storing or transferring personal data. But the requirements for both are very strict. Anonymous data is data that can not be used to identify a natural person. But the definition of anonymous data needs to be considered case-by-case, as even anonymous data can be used to indirectly identify a person.

Pseudonymisation is often used in IT systems to provide personalised services for a user. In Shibboleth the persistentID is a pseudonym of a user. This pseudonym can be passed to third parties to omit sending the real name of a natural person. This pseudonymised ID can be used, e.g., to save a reading list at a publisher for the user. The pseudonym is generated using a encrypting algorithm and the relation between natural person and pseudonym is only stored at the identity provider. This concept could be considered for services that are exclusive to a specific user, but do not need to be further adapted based on additional personal information.

VI. Conclusion and Future Work

In this paper we presented the status of research for a privacy preserving way of delivering service through a hybrid cloud environment. The paper presented the challenges and showed first directions for further research.

One big next step will be the development of a method for categorising user data and defining privacy policies as a basis for further research. This categorisation is then implemented in the prototype to evaluate the made assumptions. Furthermore, it has to be examined if pure configuration changes (e.g., user-interface-colour set to blue) and the configuration data can be

assumed as anonymous data and also lead to personalisation in the sense of AAL. The question would then be: Can there be personalisation without personal data?

REFERENCES

[1] United Nations, "World Population Ageing: 1950-2050," UN: Department of Economics and Social Affairs - Population Division, http://www.un.org/esa/population/publications/woldageing19502050/, Report, 2001.

[2] J. Grauel and A. Spellerberg, "Akzeptanz neuer Wohntechniken für ein selbstständiges Leben im Alter," in *Zeitschrift für Sozialreform*, June 2007, vol. Heft 2 Jg. 53, pp. 191–215.

[3] P. Mell and T. Grance, "The nist definition of cloud computing," National Institute of Standards and Technology, Special Publication 800-145, September 2011.

[4] D. Balfanz, M. Klein, A. Schmidt, and M. Santi, "Partizipative Entwicklung einer Middleware für AAL-Lösungen: Anforderungen und Konzept am Beispiel SOPRANO," in *GMS Medizinische Informatik, Biometrie und Epidemiologie*, vol. 4(3), http://www.egms.de/static/de/journals/mibe/2008-4/mibe000078.shtml, October 2008.

[5] M. D. Janse, "AMIGO - Ambient Intelligence for the networked home environment," Final activity report, 2008.

[6] M. Petzold, K. Kersten, and V. Arnaudov, "OSGi-based E-Health / Assisted Living," ProSyst, http://http://www.prosyst.com/fileadmin/ProSyst_Uploads/pdf_dateien/ProSyst_M2M_Healthcare_Whitepaper.pdf, Whitepaper, September 2013.

[7] R. Sadat, P. Koster, M. Mosmondor, D. Salvi, M. Girolami, V. Arnaudov, and P. Sala, "Part III: The universAAL Reference Architecture for AAL," in *Universal Open Architecture and Platform for Ambient Assisted Living*, R. Sadat, Ed. SINTEF, November 2013.

[8] J. E. Kim, G. Boulos, J. Yackovich, T. Barth, C. Beckel, and D. Mosse, "Seamless Integration of Heterogeneous Devices and Access Control in Smart Homes," in *Eighth International Conference on Intelligent Environments*, 2012.

[9] E. Ekonomou, L. Fan, W. Buchanan, and C. Thüemmler, "An Integrated Cloud-based Healthcare Infrastructure," in *Third IEEE International Conference on Cloud Computing Technology and Science*. IEEE Computer Society, 2011, pp. 532–536.

[10] A. Forkana, I. Khalil, and Z. Tari, "CoCaMAAL: A cloud-oriented context-aware middleware in ambient assisted living," in *Future Generation Computer Systems*, G. Fortino and M. Pathan, Eds., vol. 35, 2014, pp. 114–127.

[11] H. Kuijs, C. Rosencrantz, and C. Reich, "A Context-aware, Intelligent and Flexible Ambient Assisted Living Platform Architecture," in *Cloud Computing 2015: The Sixth International Conference on Cloud Computing, GRIDs and Virtualization*. IARIA, 2015.

[12] "ZAFH-AAL - Zentrum für angewandte Forschung an Hochschulen für Ambient Assisted Living," http://www.zafh-aal.de, [retrieved: 2014.07.18].

[13] "Agent Communication Language Specifications," http://www.fipa.org/repository/aclspecs.html, [retrieved: 2014.07.12].

[14] C. Fredrich, H. Kuijs, and C. Reich, "An ontology for user profile modeling in the field of ambient assisted living," in *SERVICE COMPUTATION 2014, The Sixth International Conferences on Advanced Service Computing*, A. Koschel and A. Zimmermann, Eds., vol. 5. IARIA, 2014, pp. 24–31.

[15] W. J. Gédéon, *OSGi and Apache Felix 3.0*, 1st ed. Packt Publishing, November 2010, no. 978-1-84951-138-4, ch. Using the OSGi Bundle Repository.

[16] "Deutsches Rotes Kreuz - Angebote," http://www.drk.de/angebote.html, [retrieved: 2015.09.11].

[17] "Arbeiter-Samariter-Bund Deutschland e.V." https://www.asb.de/de, [retrieved: 2015.09.11].

[18] "Caritas in Deutschland - Not sehen und handeln," http://www.caritas.de/, [retrieved: 2015.09.11].

[19] *Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data*, European Community Official Journal L 281, 1995. [Online]. Available: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML

[20] "Article 29 Working Party - European Commission," http://ec.europa.eu/justice/data-protection/article-29/index_en.htm, [retrieved: 2015.09.12].

[21] Article 29 Data Protection Working Party, "Opinion 4/2007 on the concept of personal data," *WP 136*, no. 01248/07/EN, 2007. [Online]. Available: http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2007/wp136_en.pdf

[22] "Shibboleth - What's Shibboleth?" https://shibboleth.net/about/, [retrieved: 2015.09.11].

In depth analysis of the ns-3 physical layer abstraction for WLAN systems and evaluation of its influences on network simulation results

Christopher Hepner
Institute of Communication Technology
University of Applied Sciences Ulm
Ulm, Germany
Email: hepner@hs-ulm.de

Arthur Witt
Institute of Communication Technology
University of Applied Sciences Ulm
Ulm, Germany
Email: awitt@hs-ulm.de

Roland Muenzner
Institute of Communication Technology
University of Applied Sciences Ulm
Ulm, Germany
Email: muenzner@hs-ulm.de

***Abstract*—In network simulations a correct representation of the physical layer is essential in order to achieve reliable results which are comparable to real hardware performance. The network simulator ns-3 specifies two error rate models for the calculation of the bit error rates and corresponding packet error rates for orthogonal frequency division multiplexing, the YANS [1] and NIST [2] error rate models. In [2] both models are validated and the NIST model is recommended for the calculations because of the overly optimistic results in the YANS model. However, still some inaccuracies are present in both of the models which shortly have been discussed in [3], where also a new model has been briefly sketched. In this work, the new model initially proposed in [3] is outlined in detail, including a full discussion of the differences to the prior models in [1] and [2]. Additionally, corrections for the inaccuracies in the YANS [1] and the NIST [2] models are proposed. The upper bounds for the packet success rate, calculated with the new model as well as with the corrected versions of the YANS [1] and the NIST [2] models are compared with an exact physical layer simulation using Matlab and with measurements over an AWGN channel with a typical IEEE 802.11 a/b/g/n wireless LAN module. Thereby the new model leads to results that are consistent with exact simulations of the IEEE 802.11a PHY and being comparable to real hardware performance.**
Showing the importance of a correct abstraction of the physical layer, finally the influences on network simulation results of the three models are evaluated in a ns-3 simulation covering a hidden node scenario as an example of a typical WLAN network problem.

I. Introduction

Wireless Local Area Networks (WLAN) are used in many application architectures such as Smart Home, Internet of Things, Emergency Systems or Vehicular Networks. In order to predict the behavior of such a communication network, the network can be simulated by a network simulator because of the high complexity and effort of real testbed measurements. The network simulator ns-3 is a packet-based, discrete-event network simulator with modules according to the WLAN standard IEEE 802.11 [4] and others [2]. As in most network simulators, the ns-3 physical layer implementation is abstracted. A high simulation time due to a complex implementation of the physical layer and memory expenses are the reasons. This comes with the cost that the influence of alternative physical layer techniques on the network performance is not observable. The accuracy of the simulation results is dependent on the reasonability of the abstraction. The ns-3 physical layer model is based on the calculation of bit error rates (BER) taking into account the forward error correction e.g. present in IEEE 802.11a. The model calculates the received signal–to–noise ratio (SNR) based on parameters used in the simulation model and calculates a packet error rate (PER) based on the mode of operation (e.g. modulation, coding rate) to determine the probability of successfully receiving a frame (packet success rate - PSR). Two error rate models are present in ns-3, the YANS (Yet Another Network Simulator, [1]) and NIST (National Institute of Standards and Technology, [2]) model. Assuming an Additive White Gaussian Noise channel (AWGN), binary convolutional coding and hard decision viterbi decoding, the PER can be upper bounded by using the equations in section II-B and II-C. Both models are validated and a new model which clarifies some inaccuracies is presented. Finally the calculated PSRs are validated by comparison with the results of an exact simulation of the IEEE 802.11a physical layer in Matlab and measurements of a typical WLAN module. Needed changes to the simulator are proposed and the importance of the physical layer abstraction is shown by applying all three models within network simulation of a typical hidden node scenario for a WLAN mesh network.

II. State of the Art – Validation of ns-3 error rate models

A. Abstraction of physical layer

The implementation of the ns-3 IEEE 802.11a physical layer model follows the implementation described in [1] for both, the YANS and the NIST error rate model. The physical layer is abstracted by using a path-loss propagation model in order to calculate the SNR at the receiver, followed by the calculation of a raw BER covering the used modulation and by deriving a resulting error probability (BER coded) based on the decoding of the convolutional coded bits. Finally, the PER can be derived, according to [1] by drawing a random number from a uniform distribution within the interval $[0, 1]$ and comparing it to the PER. If the random number is larger than the PER, then the packet is assumed to be successfully received. Several path-loss models are available in ns-3, such as the Log Distance Path-Loss Model or the COST-Hata Model. An overview and evaluation of these path loss models

is given in [5]. The two error rate models available in ns-3 which cover the calculation of the corresponding bit error and packet error rates are discussed in the following sections.

B. YANS error rate model

According to the YANS model [1], the BER p_{BPSK} for BPSK (Binary Phase-Shift Keying) and p_{QAM} for QAM (Quadrature Amplitude Modulation) are calculated by equations (1) - (4). Thereby the bit error rate for the M-QAM-system is calculated by transfering the M-QAM system to an equivalent $\sqrt{M}$-PAM-system (Pulse Amplitude Modulation). In this paper we have corrected equation (3) with respect to what is presented in [1] by a factor $1/\log_2(M)$ in the error probability for a single symbol in order to correctly take into account the number of bits per QAM symbol. The calculation according to equation (3) stands also in conjunction to the implementation in ns-3. $P_{\sqrt{M}}$ is the probability of error of a $\sqrt{M}$- ary PAM system with half of the average power in each quadrature signal of the equivalent QAM system. [6]

$$X = \mathrm{erfc}\left(\sqrt{\frac{1.5}{M-1}\log_2(M)\frac{E_b}{N_0}}\right) \tag{1}$$

$$P_{\sqrt{M}} = \left(1 - \frac{1}{\sqrt{M}}\right) X \tag{2}$$

$$p_{QAM} = \frac{1-(1-P_{\sqrt{M}})^2}{\log_2(M)} \tag{3}$$

$$p_{\mathrm{BPSK}} = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \tag{4}$$

The calculation of the SNR per bit in YANS is shown in equation (5). $\frac{E_b}{N_0}$ is calculated by the fraction of the noise bandwidth B_t over the raw bitrate R_b times the Signal-to-noise-plus-interference-ratio (SNIR).

$$\frac{E_b}{N_0} = \mathrm{SNIR}\frac{\mathrm{B}_t}{\mathrm{R}_b} \tag{5}$$

The performance achieved by the Viterbi decoding algorithm in the receiver can be estimated by calculating the first-event-error probability. Thereby the pairwise error probability $P_2^u(d)$ of incorrectly selecting a path within the trellis when the Hamming distance d is even or odd is shown in equation (6). p is the bit error probability p_{BPSK} for BPSK or p_{QAM} if QAM is used. [6]

$$P_2^u(d) = \begin{cases} \sum\limits_{i=\frac{d+1}{2}}^{d} \binom{d}{i} p^i (1-p)^{d-i} & (\text{d odd}) \\ \sum\limits_{i=\frac{d}{2}+1}^{d} \binom{d}{i} p^i (1-p)^{d-i} & \\ \quad + \frac{1}{2}\binom{d}{\frac{d}{2}} p^{\frac{d}{2}}(1-p)^{\frac{d}{2}} & (\text{d even}) \end{cases} \tag{6}$$

There is no simple exact expression for the first-event error probability P_e, but the error probability can be overbounded by the sum of the pairwise error probabilities $P_2^u(d)$ over all possible paths that merge with the all-zero path at the given node within the trellis diagram [6]. With this calculation scheme the so called union bound can be obtained [6] which is used in YANS [1]. The coefficients α_d represent the number of paths corresponding to the set of distances d.

$$P_e < \sum_{d=d_{free}}^{\infty} \alpha_d P_2^u(d) \tag{7}$$

The union bound (Eq. (7)) is calculated in the YANS model for the sum of $\alpha_d P_2^u(d)$ with $d = d_{free}$ (i.e. K=1) for BPSK and $[d = d_{free}, d_{free}+1]$ (i.e. K=2) for QAM (K is the number of coefficients α_d used for the calculation of the error bound). Finally, the packet success rate (PSR) is calculated by using equation (8) with the number of raw bits L.

$$PSR \leq (1-P_e)^L \tag{8}$$

C. NIST error rate model

For the calculation of the BER in the ns-3 NIST model [2] the equations (10) - (13) are used. These equations represent an approximation of equations (1) - (3) with inserted values of the number of constellation points M. Equation (10) is corresponding to equation (4) of the YANS model. The SNIR in equation (9) represents the SNR per symbol and doesn't account for the ratio of used subcarriers in the OFDM system as used in the YANS model.

$$\frac{E_b}{N_0} = \frac{\mathrm{SNIR}}{\log_2(\mathrm{M})} \tag{9}$$

$$p_{\mathrm{BPSK}} = \frac{1}{2}\mathrm{erfc}\left(\sqrt{SNIR}\right) \tag{10}$$

$$p_{\mathrm{QPSK}} = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{SNIR}{2}}\right) \tag{11}$$

$$p_{\text{16-QAM}} = \frac{3}{4\cdot 2}\mathrm{erfc}\left(\sqrt{\frac{SNIR}{5\cdot 2}}\right) \tag{12}$$

$$p_{\text{64-QAM}} = \frac{7}{8\cdot 3}\mathrm{erfc}\left(\sqrt{\frac{SNIR}{21\cdot 2}}\right) \tag{13}$$

In the ns-3 NIST model, equations (14) and (15) are used for the calculation of an upper bound (also called Chernoff Bound) P_b^c on the probability of a bit error instead of the the first-event error probability P_e in YANS. Using the upper bound in equation (14) instead of the expressions in equation (6) yields to a looser upper bound on the probability of a bit error P_b^c. In this approach it is assumed, that the probability of a bit error can be determined by using the number of nonzero information bits that are in error when an incorrect path is selected in the trellis as compared to the the all-zero path. Thereby the multiplication factors β_d correspond to the number of nonzero information bits that are in error when an incorrect path is selected for the specified hamming distance d. p is the bit error probability for the different modulations. In this way the Chernoff bound can be obtained.[1] [6]

$$P_2^c(d) < [4p(1-p)]^{d/2} \tag{14}$$

[1] b = (1, 2, 3) for code rates (1/2, 2/3, 3/4). The additional factor of $1/2$ which is presented in [2] but not in [7] is not further used.

$$P_b^c < \frac{1}{2b} \sum_{d=d_{free}}^{\infty} \beta_d P_2^c(d) \tag{15}$$

The packet success rate (PSR) is calculated as shown in the YANS model by using equation (16) with the number of raw bits L.

$$PSR \leq (1 - P_b^c)^L \tag{16}$$

The difference between the two models lies in the fact, that YANS is using the union bound with K=1 for BPSK and K=2 for QAM and NIST is using the Chernoff bound with K=10. Beside that, the SNIR is calculated in a different manner which leads to an additional shift of the resulting PSR. The SNIR in equation (9) represents the SNR per symbol and doesn't account for the ratio of used subcarriers in the OFDM system as used in the YANS model in equation (5) by the fraction of the noise bandwidth B_t over the raw bitrate R_b.

III. New ns-3 error rate model

A. New model for abstraction of the physical layer

So far the YANS model which is using the union bound (Eq. (6) and (7)) and the NIST model which is using the Chernoff bound (Eq. (14) and (15)), detailed in sections II-B and II-C, are using an inaccurate calculation of the corresponding SNR and a wrong calculation of the PSR by an inadequate number of bits L (Eq. (8) and (16)). The new model detailed in the following section is using the upper bound (Eq. (21)), a correct representation of the SNR (Eq. (20)) and a correct number of data bits for the calculation of the PSR equivalently to equation (16).

The relation between the available energy per bit E_b and the transmitted energy per OFDM symbol E_{OFDM} is described in [8]. This is the energy distributed along the whole symbol, including signaling overheads (see Eq. (17)).

$$E_{\mathrm{b}} = E_{\mathrm{OFDM}} \left(\frac{N_{\mathrm{FFT}}}{N_{\mathrm{FFT}}+N_{\mathrm{CP}}} \right) \left(\frac{N_{\mathrm{data}}}{N_{\mathrm{data}}+N_{\mathrm{pilot}}} \right) \cdot \left(\frac{1}{N_{\mathrm{data}} N_{\mathrm{BPSCS}} R} \right) \tag{17}$$

N_{FFT} is the FFT length[2]. N_{CP} is the length of the cycling prefix (CP). N_{data} is the number of data subcarriers and N_{pilot} the number of pilot subcarriers respectively. N_{BPSCS} is the number of coded bits per symbol in each OFDM subcarrier and R the code rate of the Forward Error Correction (FEC). Based on the relation in equation (17) the SNR (Signal-to-Noise Ratio) and $\frac{E_{\mathrm{b}}}{N_0}$ (SNR per bit or the ratio of energy per bit to the one-side noise spectral density N_0) can be determined. The SNR is defined as the ratio of signal power to noise power. The signal power is the energy (variance) per time sample. [8]

$$P_{\mathrm{signal}} = \frac{E_{\mathrm{OFDM}}}{N_{\mathrm{FFT}}+N_{\mathrm{CP}}} \tag{18}$$

N_0 is the noise power. Therefore using (17) and (18) the SNR can be calculated as:

$$SNR = \frac{E_{\mathrm{b}}}{N_0} \left(\frac{N_{\mathrm{data}}+N_{\mathrm{pilot}}}{N_{\mathrm{FFT}}} \right) (N_{\mathrm{BPSCS}} R) \tag{19}$$

The representation in YANS, equation (5) doesn't take into account the energy per bit to noise power spectral density ratio $\frac{E_b}{N_0}$ correctly which is used for the BER calculations. $\frac{E_b}{N_0}$ is calculated in equation (5) by the raw bit rate R_b (calculated by the number of bits per OFDM symbol) over the symbol interval time. This formulation doesn't take into account the reduction of energy due to the cycling prefix (1st term in Eq. (17)) and the reduction of the net energy due to the pilot carriers which do not transport information (2nd term in Eq. (17)) [8]. On the other hand equation (9) of the NIST model doesn't account for the ratio of used subcarriers in the OFDM system and CP at all. Equations (1) - (4) can be used in order to calculate the bit error probability p_{BPSK} for BPSK and p_{QAM} for QAM, where, however, in order to correctly calculate the raw bit error probability, the raw $\frac{E_b}{N_0}$ (Eq. (20)) has to be used. [3]

$$\left(\frac{E_{\mathrm{b}}}{N_0} \right)_{\mathrm{raw}} = SNR \left(\frac{N_{\mathrm{FFT}}}{N_{\mathrm{data}}+N_{\mathrm{pilot}}} \right) \left(\frac{1}{N_{\mathrm{BPSCS}}} \right) \tag{20}$$

The performance achieved by the Viterbi decoding algorithm in the receiver then can be estimated by calculating the upper bound P_b^u on the bit error probability. The probability of incorrectly selecting a path when the Hamming distance d is even or odd is shown in YANS equation (6). As discussed above the error probability can be overbounded by the sum of the pairwise error probabilities $P_2^u(d)$ over all possible paths that merge with the all-zero path at the given node. The multiplication factors α_d used for the calculation of the union bound in YANS correspond to the number of paths of the set of distances d. Instead the multiplication factors β_d which corresponds to the number of nonzero information bits that are in error when an incorrect path is selected for the specified hamming distance d have to be used to obtain the so called upper bound on the probability of a bit error. The coefficients and d_{free} for punctured Codes can be taken from [9]. We thus obtain the upper bound for the error probability according to equation (21). [7]

$$P_b^u < \frac{1}{b} \sum_{d=d_{free}}^{\infty} \beta_d P_2^u(d) \tag{21}$$

The packet success rate (PSR) is then calculated using equation (16) where the number of data bits L has to be used instead of the number of raw bits[4].

B. Validation of the error rate models

The upper bounds are validated using a Matlab implementation of the equations shown in sections II-B, II-C and III-A and an exact physical layer simulation of an IEEE 802.11a OFDM transceiver system (Figure 2). Figure 1 shows the results of the ns-3 implementations for the YANS and NIST model and the calculation of the union bound, upper bound and Chernoff bound for K=1 and K=10 in Matlab compared to the new model for BPSK (R = 1/2). The SNR in the Matlab implementation is calculated by using equation (20). The bit error probability p_{BPSK} is calculated as in the YANS and NIST models shown in equations (4) and (10). The union

[2]For an IEEE 802.11a OFDM system $N_{\mathrm{FFT}} = 64$ (FFT - Fast Fourier Transform), $N_{CP} = 16$ with a guard interval of 0.8μs, $N_{\mathrm{data}} = 48$, $N_{\mathrm{pilot}} = 4$, $N_{\mathrm{BPSCS}} = 6$ for QAM64, R=1/2, 2/3, 3/4.

[3]It has to be noted that interference is assumed to be zero and SNR is used in the calculations instead of SNIR.

[4]raw bits = data bits / code rate.

bound is calculated by equations (6) and (7), the upper bound is calculated by equations (6) and (21) and the Chernoff bound is calculated by equations (14) and (21) respectively. The packet success rate (PSR) for the union, upper and Chernoff bound is then calculated by using equation (16) with the number of data bits $L = 8000$ bits while YANS and NIST are calculated with the number of raw bits.

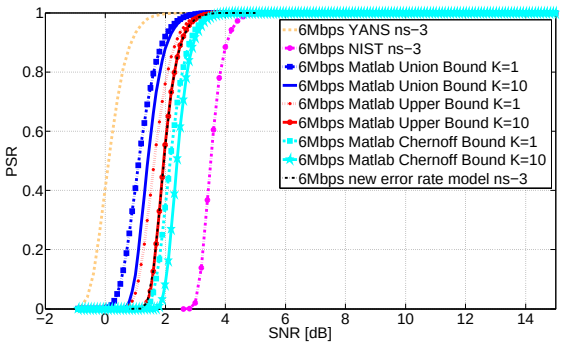


Fig. 1. Comparison of ns-3 and Matlab implementation.

As expected, the Matlab implementation of the upper bound with K=10 (red solid line) corresponds to the ns-3 implementation of the new error rate model (black dash-dot line). The upper and Chernoff bound vary less than 1dB for BPSK (R=1/2). The ns-3 YANS model is using the union bound with K=1 and the NIST model is using the Chernoff bound with K=10. Because of the inaccurate SNR calculation and the wrong number of bits for the calculation of the PSR (raw bits instead of data bits which are double the size for R=1/2) the YANS model is overly optimistic (about 2dB) whereas the NIST model is pessimistic (about 2dB) when compared to the Matlab implementations using the respective bounds used in the YANS and NIST implementations. The results outlined above indicate the following changes to the ns-3 implementation of the YANS and the NIST model: First the number of raw bits used in equation (16) must be multiplied by the code rate R. Second the calculation of the SNR must be changed to the representation in equation (20). Third it is recommended to use the upper bound for the calculation instead of the Chernoff or the union bound.

The exact simulation of the physical layer for an OFDM link according to IEEE 802.11a includes the convolutional coding, interleaver, modulation and cycling prefix at the transmitter and an ideal channel estimation, demodulation, deinterleaver and hard decision Viterbi decoder at the receiver (Fig. 2). The traceback length of the Viterbi decoder is 35 in all simulations which is 5 times the constraint length of the convolutional code with the generator polynomial [133,171]. The SNR in the simulation is calculated using equation (19). Figure 3 shows the result of the exact physical layer simulation in Matlab. The simulation has been carried out using 10^7 symbols. The result asymptotically approaches the upper bound with K=10 and confirms the calculation of the new model. Finally, the implementation of the new error rate model in ns-3, proposed in this paper and based on the calculation of the upper bound with K=10, showed a computational overhead of about 14% compared to the NIST implementation. The ns-3 simulations have been carried out by sending 81000 packets.

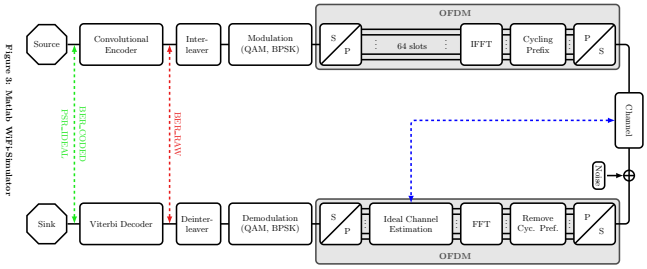


Fig. 2. Exact physical layer simulation in Matlab of an IEEE 802.11a OFDM communication link.

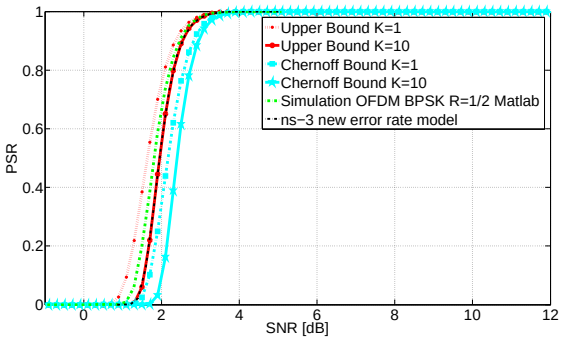


Fig. 3. Comparison of the BPSK PSR from Upper Bounds and exact Matlab simulation of the physical layer for an IEEE 802.11a communication system.

IV. Wireless Testbed Measurements

A. Estimation of hardware parameters

The PSR of a typical WLAN module over an AWGN channel is measured with a transmitter and receiver module using a wired connection with a variable attenuator. The WLAN receiver module is placed in a shielded box. The schematic of the receiver is shown in Figure 4. The module consists of an external Front-End-Module (FEM) [10] including a low noise amplifier (LNA) and a switch. The noise figure (NF) and gain of the LNA are shown in Table I. Measurements of an evaluation board of the FEM confirmed the values shown in Table I.

TABLE I. Noise Figure and Gain

Hardware Parameters	
NF LNA1	NF1 = 2.8 dB
Gain LNA1	$G1_{dB}$ = 12.0 dB
NF LNA2	NF2 = 5.8 dB
Noise Figure	NF = 3.185 dB

In order to compare the measured PSR to the simulation results, the SNR at the A/D converter in the WLAN receiver has to be estimated. One possibility is to use the Received Signal Strength Indicator (RSSI) provided by the WLAN chip (Fig. 4). According to [4], the RSSI is an optional parameter that has a value of 0 through RSSI Max. This parameter is a measure provided by the PHY sublayer and indicating the energy observed at the antenna used to receive the current Physical Protocol Data Unit (PPDU). An absolute accuracy of the RSSI values reported by the modules is not specified [4]. [11] shows a practical conversion from RSSI to dBm values. The mean RSSI from different vendors have been compared in [12] revealing significant differences between individual

vendors. The RSSI reported is also dependent on design choices made by the manufacturer. For more advanced chipsets using IEEE 802.11n multiple-input multiple-output techniques this becomes even more challenging and the calculation of the RSSI is not published by the vendors. Because of these inaccuracies the measurement of the RSSI is not practicable to evaluate the PSR. Another approach is the measurement of the signal power at the receiver input and calculation of the SNR at the ADC by subtracting the calculated noise figure of the LNAs. For this purpose, an exact knowledge of the RF

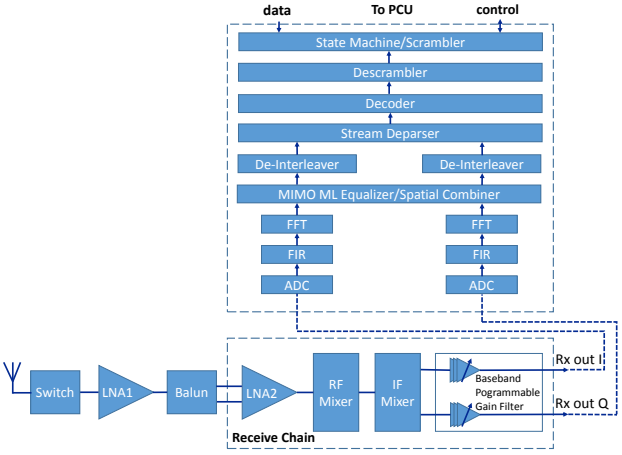


Fig. 4. WLAN receiver block.

The SNR at the A/D converter in the WLAN chip (Fig. 4) can be calculated by subtracting the thermal noise floor N_1 and the noise figure NF of all components in the signal chain from the measured received power P_{rec} at the antenna connector (Eq. (22)) (assuming quantization noise is neglectable).

$$SNR\,[\text{dB}] = P_{\text{rec}}\,[\text{dBm}] - N_1\,[\text{dBm}] - NF \tag{22}$$

With the Noise Floor N_1

$$N_1 = 10\log\left(\frac{kTB}{1\text{mW}}\right) \tag{23}$$

and the over all Noise Figure NF.

$$NF = 10\log(F) \tag{24}$$

$$F = F_1 + \frac{F_2 - 1}{G_1} \tag{25}$$

k is the Boltzman constant, T the temperature and B the noise Bandwith. F_1 and G_1 are the noise factor and gain of the external FEM (LNA1 Tab. I). F_2 is the noise factor of the internal LNA (LNA2 Tab. I). The packet size in the measurements and calculation of the PSRs is 1000 bytes.

B. AWGN channel measurements

Figure 5 shows the result of the measurement over an AWGN channel compared to the upper bound and Chernoff bound for K=1 and K=10 as well as the ns-3 implementation of the model proposed in this paper. The results show a good correspondence of the measurement results with the upper and Chernoff bound for K=10 as well as with the ns-3 implementation of the new model. With the changes described in the previous sections and including the noise figure of the WLAN hardware module in the network simulation, the YANS and NIST model – then using a correct SNR calculation and correct number of bits in ns-3 – could be used as well to estimate the performance of the physical layer of a typical WLAN module. In this case YANS would correspond to the union bound with K=1 and NIST would correspond to the Chernoff bound with K=10. With the hint that as seen in Figure 1 the YANS model changed would still show too optimistic results of about 1dB and the NIST model would show pessimistic results by less than 1dB compared to the new model.

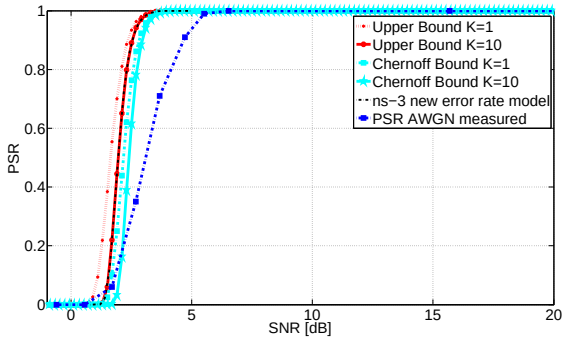


Fig. 5. Result of the ns-3 implementation of the new model when compared to the Upper Bounds and PSR of BPSK for R=1/2 over an AWGN channel.

The flattening of the measured AWGN graph might be due to inaccuracies in the measurement setup, where only the WLAN receiver module is placed in a shielded box. Part of the signal from the transmitting module might be propagating over the air and feeding back on the screen of the cable, finally entering at low amplitude the screened box where the receiver is placed. This results in a slightly frequency selective fading characteristic. To prevent this the transmitting module needs to be placed in a shielded box as well and cable lengths should be as small as possible. Therefore essential for the validation is the region of low packet success rates.

V. Network Simulation Results in ns-3

The evaluation is carried out with four communication links, involving five nodes which are placed in a rectangular grid. Figure 6 shows the topology in ns-3 (version ns3-22) where distances are in meter. Each node is initialized as a Mesh Station (MSTA) based on the IEEE 802.11s standard [4] and is part of a mesh network. Each MSTA which is in range of another MSTA establishes a peer link in between them and allows the routing of data throughout the stations. In Fig. 6 the communication links from Node1 to Node3 (Connection 1) and Node3 to Node1 (Connection 2) are relayed by Node2 because no direct link is possible. A relaying over Node4 is also possible which is decided by the routing process. Node4 and Node5 are connected by a direct link (Connection 3 and Connection 4). Also some of the nodes are hidden to others. Each communication link has a duration of 100 sec and each connection is modeled as UDP data stream with 500 kbps. Connection 1 starts at a simulation time of 2 sec, Con. 2 at 2.1 sec, Con. 3 at 10.2 sec and Con. 4 at 10.3 sec. With the

packet payload of 922 bytes, IP and UDP headers the packet becomes 1000 bytes in total. For the calculation of the received

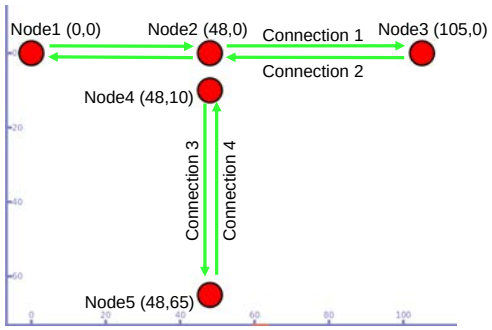


Fig. 6. Topology of the Simulation in ns-3 with 5 nodes and 4 connections.

signal power at the individual nodes, the Log Distance Path-loss model is used with an exponent of 2.7 and the Reference Loss calculated at a center frequency of 5.6 GHz. The transmit power is chosen 0 dBm and the noise figure is 3.185 dB. Tab. II shows the resulting SNR values and PSRs of the three models based on the simulation parameters for different transmission distances d_t for BPSK coderate 1/2.

TABLE II. RESULTS OF THE PSR FOR DIFFERENT DISTANCES d_t

Distance d_t	48 m	55 m	57 m	65 m
SNR	4.973 dB	3.377 dB	2.9583 dB	1.4182 dB
YANS PSR	1.0	1.0	0.9999	0.9795
NIST PSR	0.9972	0.3502	0.0157	0
New Model PSR	1.0	0.9947	0.9761	0.0277

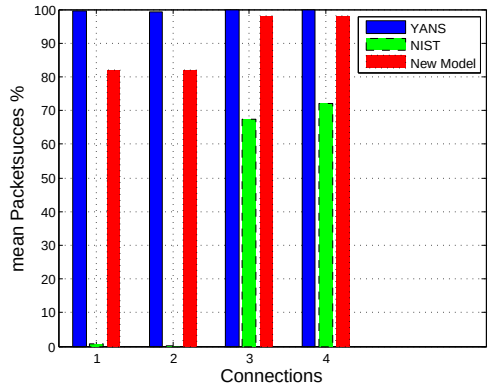


Fig. 7. Mean packetsuccess in % depending on the configuration.

Figure 7 shows the mean packetsuccess of each individual connection of the three models (YANS, NIST and the New Model) with 1000 simulation runs. The different error rate models yield to different PSR's for a given SNR, resulting in different transmission ranges as seen in Tab. II and the specified model is highly sensitive to this fact. The YANS model has a packetsuccess of almost 100% for all connections. With the NIST model almost no transmission is possible with Connection 1 and 2 and the New Model has a packetsuccess of about 82% for Connection 1,2 and 98% for Con. 3,4.

VI. CONCLUSION

The simulation results of the new ns-3 error rate model in sec. III show a good correspondence with the measurement results of a typical WLAN module (sec. IV-B) as well as with an exact physical layer simulation in Matlab of an IEEE 802.11a OFDM communication link (sec. III-B). Key elements of the new model are the usage of the upper bound (Eq. (21)) with $K \geq 10$, correctly taking into account the SNR at the receiver input (Eq. (20)) and correctly taking into account the number of data bits used for the calculation of the PSR in equation (16). Because of the inaccurate SNR and PSR calculations in YANS and NIST the YANS model is overly optimistic (about 2dB) whereas the NIST model is too pessimistic (about 2dB) when compared to the new error rate model. The calculation of the SNR should be changed in both, the ns-3 NIST and YANS error rate models in order to correctly take into account the error rate in the OFDM system. The number of data bits used in the calculation of the packet success rate should be changed as well in order to take the convolutional coding into account correctly. With the proposed changes both models could be used, resulting in more reliable performance estimations of the PSR. The Chernoff bound shows a deviation of less than 1dB when compared to the upper bound and can be used in time consuming simulations. In order to get reliable results in the ns-3 simulations, it is crucial to take into account the noise figure of the RF-Front-End. Finally it has been shown, that the network simulation results are highly sensitive to the specified error rate model which shows that an abstraction of the physical layer avoiding any unnecessary inaccuracies is key for reliable simulation results on network level.

REFERENCES

[1] M. Lacage and T. R. Henderson, "Yet Another Network Simulator," *Proceeding from the 2006 workshop on ns-2: the IP network Simulator*, 2006.

[2] G. Pei and T. R. Henderson, "Validation of OFDM error rate model in ns-3," *Boeing Research & Technology*, 2010.

[3] C. Hepner, A. Witt, and R. Muenzner, "Extended Abstract, A new ns-3 WLAN error rate model - Definition, validation of the ns-3 implementation and comparison to physical layer measurements with AWGN channel," *Workshop on ns-3 WNS3 2015, Castelldefels (Barcelona), Spain*, 2015. [Online]. Available: https://www.nsnam.org/wp-content/uploads/2015/04/WNS3_2015_submission_34.pdf

[4] IEEE Computer Society, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11*, 2012.

[5] M. Stoffers and G. Riley, "Comparing the ns-3 Propagation Models," *IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 2012.

[6] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 2001.

[7] A. J. Viterbi, "Convolutional Codes and Their Performance in Communication Systems," *IEEE Transactions on Communications Technology, Vol. Com-19, No.5*, 1971.

[8] A. M. Camara and R. P. F. Hoefel, "On the Performance of IEEE 802.11n: Analytical and Simulation Results," *XXIX Simposio Brasileiro de Telecomunicacoes - SBrt'11*, 2011.

[9] D. Haccoun and G. Begin, "High-Rate Punctured Convolutional Codes for Viterbi and Sequential Decoding," *IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 37, NO. 11*, 1989.

[10] SKYWORKS, "SE5516A: Dual-Band 802.11a/b/g/n/ac WLAN Front-End Module," *DATA SHEET*, 2013.

[11] J. Bardwell, "Converting Signal Strength Percentage to dBm Values," *Wild Packets Application Note*, 2002.

[12] G. Lui, T. Gallagher, B. Li, A. G. Dempster, and C. Rizos, "Differences in RSSI Readings Made by Different Wi-Fi Chipsets: A Limitation of WLAN Localization," *IEEE International Conference on Localization and GNSS (ICL-GNSS)*, 2011.

Requirements Analysis for Privacy-Protecting Solutions

Florian Kemmer, Christoph Reich, Martin Knahl
Cloud Research Lab
Furtwangen University
Furtwangen, Germany
{Florian.Kemmer,Christoph.Reich,
Martin.Knahl}@hs-furtwangen.de

Nathan Clarke
Centre for Security, Communications and Network Research
University of Plymouth
Plymouth, United Kingdom
N.Clarke@plymouth.ac.uk

***Abstract*—Cloud Computing is an increasingly popular paradigm that allows instant access to virtually unlimited resources via the Internet. Despite all research efforts during the past years, the security state, however, remains uncertain and source of mistrust and problems. These problems amplify with the advent of more pervasive technology like Mobile Cloud Computing, where more and more end users begin utilising these services – often ineptly or even unknowingly with unintended consequences to their privacy. The purpose of this paper is to briefly outline the general problematic in Cloud Computing privacy, present various existing solutions to preserve privacy in such an environment and describe a feature set that an end user facing tool would need to provide in order to efficiently and effectively allow control over sensitive data.**

I. Introduction

Cloud Computing is a paradigm that allows instant on-demand access to virtually unlimited resources of the Cloud Service Provider (CSP) over the internet. Having access to this vast computing power while only renting, but not buying the necessary hardware grants the flexibility to up- and downscale at will and has lead to a wide adoption of this concept.

Despite that widespread usage, security issues are far from being solved, both psychologically as well as technologically [1]. Most common source of security considerations is the shared nature in public clouds, where potentially sensitive data is stored and processed not only not exclusively, but right next to virtual machines of competitors [2]. It all boils down to the fact that, once data leaves corporate boundaries, it also escapes the company's control.

It only seems consequent, that fear of data loss or data breaches still remain among the most often mentioned fears [3] and prevent the Cloud Computing paradigm from fulfilling its potential [4]. While the technological fundamentals are well understood, the security state still remains "puzzling" [5].

The aforementioned problems get worse when end users are exposed to cloud services, which is increasingly happening in today's world driven by more and more pervasive technology [6].

Parallel to the rise of Cloud Computing, smart devices have become increasingly popular, forming the field of Mobile Cloud Computing, which can be defined as "an integration of cloud computing technology with mobile devices to make the mobile devices resource-full in terms of computational power, memory, storage, energy, and context awareness" [7].

While it only seems logical to combine resource-limited smart devices with infinite computing power from the cloud, it also makes cloud services almost omnipresent, integrated into daily lives and easier to access and harder to identify as such. This isn't necessarily bad, but it also makes inadvertent exposure of personal sensitive data a more imminent problem. Not only are personal users often less educated with regards to technology than company professionals, but also simply less aware or less caring about the risks involved. Finding ways to adequately protect users' data and privacy is thus more required than ever. It's easier than ever to unintentionally share sensitive information, which now not only contain some secret company numbers, but potentially health-related information about the user itself.

The rest of the paper is structured as follows: Section II attempts to find a definition of the concept "privacy" and describes end-users' attitude as well as threats towards it. In section III, we will briefly show, why cryptography can only be (an important) part of the solution but by itself is not enough to protect sensitive data in cloud environments. Consequently, section IV will present and analyse existing privacy-protecting approaches to devise requirements imposed on such solutions. Section V will draw a conclusion and summarise the findings.

II. End-users and their privacy

Similarly to the corporate environment, end-users are afraid of potential misuse of their information and worried about who has access, while still using the services anyway [8], [9]. In contrast to companies, they're often not as well educated in regards of computer security and thus unable to react appropriately or sometimes even unaware of their utilisation of cloud services when tightly integrated into other systems [7].

Moreover, users traditionally show risky online behaviour and are unwilling or unable to change their behaviour [9], [10].

A. Definition of Privacy

Many authors have tried to come up with a holistic definition of the concept of privacy and came to the conclusion that it's far from an easy task. As an interdisciplinary subject, authors often tend to focus a definition on their area of research while neglecting the others [11], [12], [2], [13]. Furthermore, [14] emphasised, that any definition can only be valid for a group of people, but not generally applicable as the understanding of the term varies too much depending on cultural background, personal history and other factors.

Like many authors before and backed by the survey conducted in [14], for the rest of this paper, we will fall back to a definition that puts data into focus: Privacy in our understanding is the right to know where and why personally identifiable information are collected and in particular the ability to remain in control over this data (i.e. allow and disallow access).

To demonstrate the weakness of this definition, consider the following example by [15] (based on [16]): Plenty of CCTV cameras are installed in a place. Is the privacy of people passing this place invaded? What, if these cameras are installed, but never turned on? Is the privacy still invaded? In the second scenario, no data has been collected, so by the above introduced definition, no invasion has taken place. However, people are likely to have behaved differently due to the anticipation of being observed, so obviously *something* must have happened.

B. Threats to Privacy

Two potential issues with outsourced data have been pointed out by [2]. The CSP could use entrusted data and reuse them (against the user's knowing and will) to generate profit. The concept of "data for service" is known and agreed in many areas (e.g. targeted advertising), but the potential of secondary usage, like selling to third parties, remains.

Another point made by [2] is the problematic arising from data storage in different locations and jurisdictions. Large CSPs usually have data centres across the globe and might store user data in countries with weaker privacy protection laws than the user's home country. Regulations like the *Safe Harbour Agreement* between the EU and the US were meant to ease these tensions, but have failed to solve all problems [17].

Obviously, in shared infrastructures, it always remains possible that other users are able to gain access to sensitive data, e.g. by abusing security vulnerabilities. This, however, is not strictly speaking a privacy problem, but goes back to computer security in general and is thus not discussed in detail.

With the increased amount of data collected (Social Networks, Smart Devices, Online Tracking) and more sophisticated data analysis techniques, it becomes possible to combine small and seemingly anonymous and innocent pieces of data and generate user profiles [18].

One last threat to a user's privacy is the user himself: Despite all efforts and supportive tools, the user is still able to make stupid decisions.

III. Why cryptography is not enough

Naturally, when trying to protect sensitive data in potentially hostile environments, cryptography plays an important role. Various laws and regulations require the CSP to encrypt sensitive or personally identifiable information before storing them, which protects the CSP from law enforcement, but not user data from the CSP [19].

The concept of Cloud Computing, is far too flexible to have all eventualities covered by a single tool. Simply encrypting data might be sufficient for rudimentary scenarios where a single client stores files on cloud services, such as Dropbox[1]. Assuming no technical flaw in the encryption software, this prevents the Cloud Service Provider from snooping through the user's data. The limitations of this solution become apparent when considering data sharing with other users ([20]) or the desire to not only *store* data remotely, but also use the available *computing power*.

Considering the latter point, *homomorphic encryption* as first suggested by [21] has long been seen as the "holy grail" [22] and solution to most of the problems. Fully homomorphic encryption, such as designed by [23], allows arbitrary computation over encrypted data. A remote system would thus not need to decrypt data to be able to answer user queries. While this solves a few problems, [22] point out that it is still not enough, as it by itself only satisfies the "Private, single-client" use-case. The problem of easily sharing subsets data with a changing group of people is not solved by this.

IV. Existing approaches for privacy protection and requirement analysis

In this section, we define three phases of a life cycle, sensitive data passes and introduce existing solutions to protect data during each phase. Based on these approaches, a list of requirements for privacy-protecting tools will be compiled.

A. Generation of data and publication

The best and technically easiest way to protect data from being misused it to put as little as possible out there [24]. While this statement is generally true, it suffers from similar drawbacks as mindless encryption, i.e. making sharing of data impossible. Moreover, it is often not the user's concious decision to share data, e.g. when referring to CCTV cameras or tracking of online behaviour. In those scenarios, data is automatically being collected and the user would have to actively prevent that. To do so, however, he first needs to be aware *and* bothered[2] by this.

Furthermore, collecting users' data is not always necessarily a bad thing: Personalised systems who "know" the user, his preferences and current situation often come up with better results. Searches for "restaurants near me" would not be possible without the service knowing where the user currently is.

It is thus not about simply blocking all data from being transferred to the provider. Instead, it is about selectively

[1] https://www.dropbox.com

[2] "I have nothing to hide"

reducing the amount of information shared and only expose them in return for an actual benefit.

When the user finally decides to share data, it still can be protected to some extent [25], [26], [24]. Sticking with the above example, reducing the accuracy of is location would be one option. An alternative would be transmitting various different values and receive results for multiple completely different locations. The added benefit of keeping his exact location secret, in return, leads to less optimal results. Furthermore, [27] have pointed out that this additional noise generated also results in additional overhead and thus finally in additional costs.

In corporate areas, inadvertent leakage of sensitive has been in the focus of research for a while and resulted in systems like *Data Leak Detection as a Service* by [28]. This detection, unfortunately, is not so simple to automate in personal environments, as [29] have shown at the example of smart phones: A user's perception of what's too sensitive to share and what not remains too volatile to be reflected by a strict set of rules or a set of fingerprints of sensitive data. It is thus difficult to distinguish between the inadvertent transmission of sensitive data and the deliberate decision to do so.

Nonetheless, [13] and [30] have proposed similar models to regulate data flow, similar to the above mentioned "Data leak detection", albeit being either very generic or focusing on specific problems. Special *Privacy Enforcement Points* are introduced in various locations, such as a home user's network gateway to monitor passing data. According to various rules and depending on the destination, the information flow can be blocked to ensure privacy. In both cases, however, the problem described by [29] remains: While certain rules may be consistent over time, others could be too volatile for such a system.

B. Access control and monitoring

Once the data is shared with a third party, it is important the data owner remains able to define and enforce access rights. In the most extreme scenario, this would allow him to exclude the service provider itself from being allowed to access the stored information.

As stated in the definition, privacy is perceived very differently depending on various factors, including time. Consequently, changing access rights must be possible at any time.

One way to cryptographically enforce access control is *Attribute Based Encryption*, which finds application in log encryption [20] or in various schemes to protect personal health records (PHR) [31], [32]. Using such systems allows to issue different keys to different persons who would then be able to only decrypt certain fields of a log file or a PHR. The solution of [31] also defined "break-glass" attribute to make PHRs available to emergency services when required. In such scenarios, most users are accepting a loss of privacy in order to maintain their physical health [13]

Another aspect of Access Control has been shown by [1]: Apart from restricting access, it is also important to monitor who (if multiple people/institutions are allowed) accesses the information when and why. The goal of this is to create transparency and allow the user to trace their data and thus ultimately generate trust in the system.

C. Deletion

At some point, a user might want to remove his data for one of many conceivable reasons, e.g. a change of CSP, no further need for the data or simply an uneasy feeling with certain data store online.

If the data was cryptographically secured and the key available to only one person (cf. the Dropbox example above), data deletion is of no concern for the user as it's not accessible for other entities anyway.

However, it has been shown, that this is not always the case as cryptography does not provide a holistic solution to the problems presented. It is thus to be assumed, that parts of the data remain either in plain text or decryptable by other persons. In a world of globally-distributed data centres, copies and backups of the data are equally scattered around the globe ([2]), which makes it difficult to track down every copy. Considerable amount of research has dealt with verifying the existence/availability of data, but a lot less effort has been put into veritably deleting information [33].

D. Requirements summary

Any solution that aims to protect user privacy has to fulfill various requirements and goals in order to be accepted, useful and efficient.

- *Configurability*
  Privacy is a very personal concept and consequently interpreted differently by everyone. The user must be able to modify the system to reflect his attitudes rather than being dictated what's best for him [34], [24], [32].
- *Simplicity*
  Studies have shown that the average user is not willing to invest much time; even for his own privacy [9]. A too hard to understand tool would most likely be rejected by the intended audience. In order to remain simplicity, it has been suggested to add no more than three options for any given category as is too difficult to sensibly distinguish between more choices [35]. Such options could be from "trusted/semi-trusted/untrusted" or "primary practitioner/other practitioners/other" [36].
- *Reducing data exposure*
  Whenever possible, the user should be encouraged and supported to not share information, unless absolutely required as this remains the safest form of misuse prevention [34], [13], [30]
- *User feedback and education*
  The user needs to be informed about why his data is requested, what it is going to be used for and how it's about to be processed in order to come to a sensible decision.
- *Dynamic Access Control*
  It has been shown, that the set of people allowed to

access certain data is changing and potentially frequently changing. “Break glass”-functionality as shown by [31] can proof life saving in emergencies.

- *Transparency*
  A lack of transparency creates fear and thus discourages from usage [13]. This not only means that the tools should aim for making existing processes more transparent and comprehensible ([1]), but also that they themselves must be doing so transparently.
- *Secure infrastructure*
  Obviously, the solution itself and potentially required infrastructure should implement known best practices in order to not itself become an additional threat to the user’s privacy [1].

V. Conclusion

One of the biggest problems when dealing with end-user privacy is the unclear definition of it due to its flexibility, interdependence on many other aspects, varying perception over time and interdisciplinary nature. The common view on privacy as sum of and control over personally identifiable information scores with its simplicity but can be shown to be too simple and not fully covering all aspects easily. In this paper, we have split a data life cycle into three distinct phases, analysed existing solutions to protect data during each of these phases and shown that a combination of protective measures is required to cover the whole cycle. Based on the characteristics of these approaches, we have compiled a list of requirements that any future tool would need to fulfill in order to be usable and efficient aid in the hand of end-users to enable them to protect their own privacy according to their very own definition of the term.

References

[1] M. Shabalala, P. Tarwireyi, and M. Adigun, “Privacy monitoring framework for enhancing transparency in cloud computing,” in *2014 IEEE 6th International Conference on Adaptive Science Technology (ICAST)*, Oct. 2014, pp. 1–7.

[2] S. Pearson and A. Benameur, “Privacy, Security and Trust Issues Arising from Cloud Computing,” in *2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, Nov. 2010, pp. 693–702.

[3] Top Threats Working Group, “The notorious nine: cloud computing top threats in 2013,” *Cloud Security Alliance*, 2013.

[4] D. Chen and H. Zhao, “Data Security and Privacy Protection Issues in Cloud Computing,” in *2012 International Conference on Computer Science and Electronics Engineering (ICCSEE)*, vol. 1, Mar. 2012, pp. 647–651.

[5] D. A. B. Fernandes, L. F. B. Soares, J. V. Gomes, M. M. Freire, and P. R. M. Inácio, “Security issues in cloud environments: a survey,” *International Journal of Information Security*, vol. 13, no. 2, pp. 113–170, Apr. 2014. [Online]. Available: http://link.springer.com/10.1007/s10207-013-0208-7

[6] S. Pearson, “On the Relationship between the Different Methods to Address Privacy Issues in the Cloud,” in *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, ser. Lecture Notes in Computer Science, R. Meersman, H. Panetto, T. Dillon, J. Eder, Z. Bellahsene, N. Ritter, P. D. Leenheer, and D. Dou, Eds. Springer Berlin Heidelberg, 2013, no. 8185, pp. 414–433. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-41030-7_30

[7] A. Khan, M. Othman, S. Madani, and S. Khan, “A Survey of Mobile Cloud Computing Application Models,” *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 393–413, 2014.

[8] M. Sato, “Personal data in the cloud: A global survey of consumer attitudes,” 2010.

[9] A. Horvath and R. Agrawal, “Trust in cloud computing,” in *Southeast-Con 2015*, Apr. 2015, pp. 1–8.

[10] A. A. Abdulwahid, N. Clarke, I. Stengel, S. Furnell, and C. Reich, “Security, Privacy and Usability – A Survey of Users’ Perceptions and Attitudes,” in *Trust, Privacy and Security in Digital Business*, ser. Lecture Notes in Computer Science, S. Fischer-Hübner, C. Lambrinoudakis, and J. López, Eds. Springer International Publishing, Sep. 2015, no. 9264, pp. 153–168. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-22906-5_12

[11] F. Schaub, B. Konings, and M. Weber, “Context-Adaptive Privacy: Leveraging Context Awareness to Support Privacy Decision Making,” *IEEE Pervasive Computing*, vol. 14, no. 1, pp. 34–43, Jan. 2015.

[12] A. Morton and M. A. Sasse, “Privacy is a Process, Not a PET: A Theory for Effective Privacy Practice,” in *Proceedings of the 2012 Workshop on New Security Paradigms*, ser. NSPW ’12. New York, NY, USA: ACM, 2012, pp. 87–104. [Online]. Available: http://doi.acm.org/10.1145/2413296.2413305

[13] M. Henze, L. Hermerschmidt, D. Kerpen, R. Haussling, B. Rumpe, and K. Wehrle, “User-Driven Privacy Enforcement for Cloud-Based Services in the Internet of Things,” in *2014 International Conference on Future Internet of Things and Cloud (FiCloud)*, Aug. 2014, pp. 191–196.

[14] H.-Y. Huang and M. Bashir, “Is privacy a human right? An empirical examination in a global context,” in *2015 13th Annual Conference on Privacy, Security and Trust (PST)*, Jul. 2015, pp. 77–84.

[15] A. Narayanan, “Privacy is not Access Control (But then what is it?),” Feb. 2010. [Online]. Available: http://33bits.org/2010/02/13/privacy-is-not-access-control/

[16] R. Calo, “People can be so fake: A new dimension to privacy and technology scholarship,” *Penn St. L. Rev.*, vol. 114, p. 809, 2009. [Online]. Available: http://heinonlinebackup.com/hol-cgi-bin/get_pdf.cgi?handle=hein.journals/dlr114\&section=26

[17] A. Busch, “From Safe Harbour to the Rough Sea? Privacy Disputes across the Atlantic,” *SCRIPT-ed*, vol. 3, no. 4, pp. 304–321, Dec. 2006. [Online]. Available: http://www.law.ed.ac.uk/ahrc/script-ed/vol3-4/busch.asp

[18] M. Backes, P. Berrang, and P. Manoharan, “How well do you blend into the crowd? - d-convergence: A novel paradigm for quantifying privacy in the age of Big-Data,” *CoRR*, vol. abs/1502.03346, 2015. [Online]. Available: http://arxiv.org/abs/1502.03346

[19] O. Wenge, U. Lampe, A. Müller, and R. Schaarschmidt, “Data Privacy in Cloud Computing–An Empirical Study in the Financial Industry,” 2014. [Online]. Available: http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1166\&context=amcis2014

[20] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data.” ACM Press, 2006, p. 89. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1180405.1180418

[21] R. L. Rivest, L. Adleman, and M. L. Dertouzos, “On data banks and privacy homomorphisms,” *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978. [Online]. Available: https://people.csail.mit.edu/rivest/pubs/RAD78.pdf

[22] M. Van Dijk and A. Juels, “On the Impossibility of Cryptography Alone for Privacy-Preserving Cloud Computing.” *HotSec*, vol. 10, pp. 1–8, 2010. [Online]. Available: http://static.usenix.org/events/hotsec10/tech/full_papers/vanDijk.pdf

[23] C. Gentry, “A fully homomorphic encryption scheme,” Ph.D. dissertation, Stanford University, 2009. [Online]. Available: http://cs.au.dk/~stm/local-cache/gentry-thesis.pdf

[24] M. Mowbray and S. Pearson, “A Client-based Privacy Manager for Cloud Computing,” in *Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE*, ser. COMSWARE ’09. New York, NY, USA: ACM, 2009, pp. 5:1–5:8. [Online]. Available: http://doi.acm.org/10.1145/1621890.1621897

[25] L. Sweeney, “k-anonymity: A model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0218488502001648

[26] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, “L-diversity: Privacy Beyond K-anonymity,” *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, Mar. 2007. [Online]. Available: http://doi.acm.org/10.1145/1217299.1217302

[27] G. Zhang, Y. Yang, and J. Chen, "A historical probability based noise generation strategy for privacy protection in cloud computing," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1374–1381, Sep. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022000011001656

[28] X. Shu and D. D. Yao, "Data Leak Detection as a Service," in *Security and Privacy in Communication Networks*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, A. D. Keromytis and R. D. Pietro, Eds. Springer Berlin Heidelberg, 2013, no. 106, pp. 222–240. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-36883-7_14

[29] Z. Yang, M. Yang, Y. Zhang, G. Gu, P. Ning, and X. S. Wang, "AppIntent: Analyzing Sensitive Data Transmission in Android for Privacy Leakage Detection," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13. New York, NY, USA: ACM, 2013, pp. 1043–1054. [Online]. Available: http://doi.acm.org/10.1145/2508859.2516676

[30] T. Hayakawa and T. Hikita, "Proposal for Easily Detachable Proxy for Personal Information Leakage Detection," in *Computer Science and its Applications*, ser. Lecture Notes in Electrical Engineering, J. J. J. H. Park, I. Stojmenovic, H. Y. Jeong, and G. Yi, Eds. Springer Berlin Heidelberg, 2015, no. 330, pp. 21–27. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-662-45402-2_4

[31] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, Jan. 2013.

[32] K. P. Kulkarni and A. M. Dixit, "Privacy Preserving Health Record System in Cloud Computing using Attribute based Encryption," *International Journal of Computer Applications*, vol. 122, no. 18, 2015. [Online]. Available: http://search.proquest.com/openview/50e59b31131275360209l883eb6d92ef/1?pq-origsite=gscholar

[33] Z. Mo, Q. Xiao, Y. Zhou, and S. Chen, "On Deletion of Outsourced Data in Cloud Computing." IEEE, Jun. 2014, pp. 344–351. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6973760

[34] W. Itani, A. Kayssi, and A. Chehab, "Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures," in *Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009. DASC '09*, Dec. 2009, pp. 711–716.

[35] I.-H. Chuang, S.-H. Li, K.-C. Huang, and Y.-H. Kuo, "An effective privacy protection scheme for cloud computing," in *2011 13th International Conference on Advanced Communication Technology (ICACT)*, Feb. 2011, pp. 260–265.

[36] J. Zhou, Z. Cao, X. Dong, and X. Lin, "TR-MABE: White-box traceable and revocable multi-authority attribute-based encryption and its applications to multi-level privacy-preserving e-healthcare cloud computing systems," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2015, pp. 2398–2406.

A Taxonomy for HPC-aware Cloud Computing

Holger Gantikow
science + computing ag
Tübingen
Email: gantikow@gmail.com

Christoph Reich
and Martin Knahl
Hochschule Furtwangen
Furtwangen
Email: Christoph.Reich@hs-furtwangen.de
Email: Martin.Knahl@hs-furtwangen.de

Nathan Clarke
Plymouth University
Plymouth, UK
Email: N.Clarke@plymouth.ac.uk

***Abstract*—Using Cloud offerings for resource intense High Performance Computing (HPC) seems like a self-evident step, as they offer on-demand access to virtually infinite resources. This should seem especially attractive for small and medium-sized enterprises (SMEs), usually not operating own data centers, avoiding the high investment costs that go with them.**

There are several reasons behind the fact that HPC in the Cloud is not that widespread used as one would expect. Lack of trust and fear of competitors easily gaining access to confidential data, as well as more technical reasons, like performance degradation due to widespread use of virtualization or complicating existing workflows, among them.

This paper presents a taxonomy for HPC-aware Clouds, in how far they distinguish from current offerings and how much effort it will take to raise HPC-awareness of a current cloud.

I. INTRODUCTION

In recent years the adoption of *Cloud Computing* has become increasingly popular for workloads of all kinds. As this paradigm offers convenient on-demand access to configurable resources, including services, storage and systems, it became the delivery model of choice for all sorts of use-cases.

The promise of elasticity, access to virtually infinite resources, combined with a pay-as-you-go model seems like a perfect match for resource-intense computing needs. But High Performance Computing (HPC) in cloud, often labeled *High Performance Computing as a Service (HPCaaS)*, is not that widespread used as one would expect and is only a small fraction of the business which arose around Cloud Computing.

A. HPC in Clouds

Applications in the domain of HPC have massive requirements when it comes to resources like CPU, memory, I/O throughput and interconnects. This is the reason why they are traditionally run in a bare-metal setup, directly on physical systems, which are interconnected to so-called clusters. The operating system of choice is in general Linux.

Large enterprises and scientific institutions utilize their own clusters and have HPC-specialized operators. As these clusters are usually optimized for a specific workload, the primary reason for considering a cloud-based HPC setup is usually cost-driven, especially when additional resources are only needed for a short amount of time. As their HPC data centres are frequently not yet operated as a *Private Cloud*, such *cloudbursts* still require certain work in advance and are thus seldom used. Furthermore, their systems often offer potential for optimization of existing resources, for example turning them into private clouds.

But especially for small and medium-sized enterprises (SMEs), for example in the field of computer aided engineering (CAE), the use of cloud resources for HPC seems attractive, as it avoids the significant investment costs that come with local clusters, their infrastructural needs (cooling, high-speed interconnects) and costs of operation. As they are usually short on HPC-trained IT-staff, moving from workstations to the Cloud for shorter turnaround times of compute jobs and thus increased productivity would seem natural, even though the aspect of workflow integration still would need to be solved.

B. Reasons for Limited Acceptance of HPC in Clouds

Even though projects like *The UberCloud Experiment* [UberCloud, 2015], trying to bring together cloud providers, HPC experts, independent software vendors (ISVs) and HPC users, took place over the last years, HPCaaS is still not widely used. The reasons behind this and what distinguishes HPC-aware clouds from regular clouds will be discussed in this paper, thus hopefully leading to improvements in current cloud architectures and leading to a higher amount and acceptance of HPC-aware offerings. We consider this information valuable for both private and public cloud providers, as well as cloud customers.

II. TAXONOMY OF HPC-AWARE CLOUDS

Our taxonomy is split into six main categories relevant for HPC-aware clouds, as pictured in figure 1.

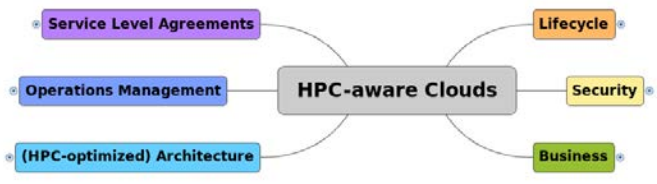


Fig. 1. HPC-aware Clouds Taxonomy: Overview of Categories

The main categories, including a brief description, are as follows:

Lifecycle
Most of the *lifecycle* aspects need adjustment on the cloud consumer side, as the use of dynamic cloud resources has a rather high impact on existing HPC workflows.

Security
The fear of *security* degradation is one of the major aspects limiting the adoption of cloud services for many potential consumers.

Business
The *business* side of a cloud offering requires some adjustments, as several additional aspects need to be included for HPC-awareness.

HPC-optimized Architecture
HPC relies on systems with a very good performance. Modifying cloud installations with focus on *HPC-optimized Architecture* is a pivotal aspect to consider.

Operations Management
There is few difference when it comes to monitoring and managing HPC-aware cloud resources. Most of the *Operations Management* adaptations need to take place on the consumer side, ensuring the suitability of obtained services.

Service Level Agreements
Extending currently established *Service Level Agreements* (SLAs) for HPC requirements is essential, as there are several aspects not covered with regular cloud SLAs.

For this paper we focused on the categories and several subcategories we identified to be most relevant for raising the HPC-awareness of a cloud offering. This includes the main categories *HPC-optimized Architecture* and *Service Level Agreements*, as well as the subcategories *HPC Add-ons* and *Resources* (both belonging to the category *Business*) and *Stability* (subcategories of the category *Security*). The subcategories are color-coded according to their affiliation to the main categories from figure 1 and the captions of the figures are also labeled accordingly. The categories *Operations Management* and *Lifecycle* are not in the focus of this paper.

The complete taxonomy, also including aspects relevant for regular clouds can be found online [Lab, 2015] at *The Cloud Research Lab (Institut für Cloud Computing und IT-Sicherheit)* which is part of the Department of Computer Science at the Hochschule Furtwangen University, Germany.

A. HPC Add-ons (Subcategory of Business)

There are several *HPC add-ons* providers need to include in their business offering for HPC-aware clouds, as pictured in figure 2.

HPC jobs in general do not only consist of one single computation process, but of several interlinked steps, including pre- and post-processing for example. Depending on the offered service model the way *workflow integration* has to be provided is different, but essential for the suitability of such a solution. When a consumer uses HPC in a cloud based

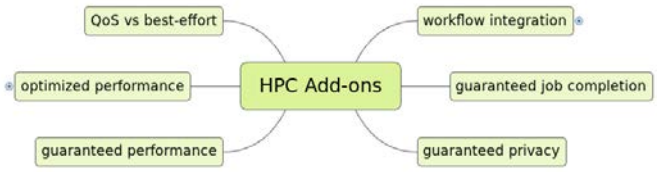


Fig. 2. HPC-aware Clouds Taxonomy Detail: HPC Add-ons (Subcategory of *Business*)

on *Infrastructure as a Service* (IaaS), means to integrate the dynamic resources into his his current load sharing mechanism, like queuing systems, are important. When utilizing a *Platform as a Service* (PaaS) resources scenario, job workflow integration is still required, as the availability of the input data for the computation has to ensured, as well as licenses - often required for commercial applications. Even though *Software as a Service* (SaaS) HPC-offerings are virtually non-existent yet, they would still require for example some sort of remote visualization tool, offering decent 3D-performance. As it is often not feasible to download output data locally due to size, one would rather decide to store all the data at the cloud provider's site, but would still require graphical access for example for 3D-modelling.

These integration aspects show that for HPC in clouds providers and customers in general are more closely tied and need to interact more than in a traditional complete self-service cloud scenario, a step providers have to be aware of.

Another important aspect is the availability of several *guarantees*, in general agreed upon in corresponding SLAs. As HPC jobs sometimes take several days to process data, a customer could require a *guaranteed job completion* policy for long-running jobs, to ensure that his computation does not get terminated prematurely, as some applications offer no checkpointing, where intermediate data gets saved, enabling a restart at the latest saved state, saving time for re-computation.

Even though a customer should be able to consider his cloud provider *trustworthy* after all, the availability of *guaranteed privacy* is important for clients processing confidential data, as they often fear privacy breaches and competitors gaining access to their data, especially in multi-tenant environments. Non-shared networking-components or customer separation on a per-system basis, preventing hypervisor-level attacks in virtualized environments, might be agreeable here for different level of guaranteed isolation.

The same conditions apply for performance, as especially customers with own data-centres fear performance degradation in a cloud scenario. Providing strictly separated components for *guaranteed performance* offerings and avoiding over-optimizing utilization by over-booking for customers requiring *optimized performance* might be valuable here. Traditional cloud providers often over-optimize their systems to a degree of capacity utilisation which makes them unsuitable for HPC applications. As far as possible a provider should offer his resources for HPC on a Quality of Service (QoS) basis rather than best-effort, as this significantly increases the service

dependability.

B. Resources (Subcategory of Business*)*

Another subcategory of the business aspects are the resources a cloud provider has to provide for a well-suited HPC cloud offering. We pictured the ones most relevant in figure 3.

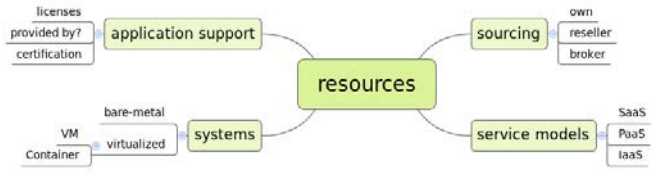


Fig. 3. HPC-aware Clouds Taxonomy Detail: Resources (Subcategory of *Business*)

When it comes to *sourcing* a provider has two options: offering access to self-hosted systems and services, or acting as a reseller of resources of a third party, sometimes with added value. Combining the two sourcing strategies in a transparent way, for example by providing long term storage from a different provider, is also possible. Also of interest could be to act as a specialized broker of HPC-aware resources, but this option is not further discussed here.

Choosing the right service model for HPC in clouds is important. Currently most providers focus on offering HPC IaaS resources, as this is the easiest way to implement for a provider. But for a consumer this is the most tasking to utilize in terms of integration. The beforehand mentioned *The UberCloud Experiment* [UberCloud, 2015] for example relied on experienced system integrators (and ISVs) for bridging the gap between consumers and providers and turn mere IaaS resources into a usable platform. Especially for SMEs with few well-versed integration specialists this delivery model is suboptimal. We consider HPC provided as PaaS, or even as SaaS as far more suitable for those customers, as the reduced integration cost would lower adoption barriers.

As further discussed in section II-C the way the underlying systems are provided is relevant to overall performance. As cloud is usually centred around virtualization as main pillar, which can decrease performance, offerings for bare-metal HPC clouds still exist, for example *Bull extreme factory* [Bull, 2015] or *Penguin Computing on Demand (POD* [Penguin Computing, 2015]. Even though they offer a cloud-like pay-per-use billing model and web portal interfaces they differ for instance from regular clouds in terms of flexibility. For example availability of applications is limited to a set of pre-configured applications, which is valuable for some consumers, but can pose a challenge if a different version or application is required as they will have to be installed by or at least with the help of the provider. The effort and time to deliver a new application can be significantly reduced by deploying applications using container-based virtualization, as it encapsulates an application and all required libraries into a single container. This is especially useful for applications with conflicting dependencies, as the underlying operating system (OS) is turned into a runtime environment for these containers, reducing the need to modify the installed OS for different applications and customers.

The aspect of *application support* is closely linked to the applied service model, as who is responsible for roll-out and configuration depends on it. This is gaining further importance if commercial software has to be used. It often requires licenses and sometimes also certification on operating system level. This usually limits the operating system choice to an enterprise Linux distribution, requiring a license or some form of subscription too. Application licensing in cloud scenarios is a complex topic, as most ISVs do not offer pay-per-use licensing mechanism. This means a user has to purchases the right to use a certain software for a maximum number of concurrent systems for an amount of time in advance, a year for example. As these licenses are usually quite expensive they are ordered corresponding to estimated regular utilization and their agreed upon usage is enforced by license servers residing on the customer site. Even when these servers can be made reachable from a cloud, this means that for a cloud-burst scenario additional licenses would need to be available, which is generally not the case, due to costs. As such dynamic usage in clouds is strongly limited by the static good software licenses, we hope for ISVs to open up and offer pay-per-use licenses, realizing this step could lead to more revenue instead of less as they fear and making it easier for cloud providers supporting commercial HPC applications.

C. (HPC-optimized) Architecture

Offering HPC-aware clouds has by far most impact on architectural aspects as how to design a cloud environment and what sort of systems are required. Figure 4 shows what infrastructural optimizations should be considered for HPC-awareness.

From a basic hardware *systems* perspective applications in the field of HPC rely on ideally *latest generation hardware*, as they provide optimal performance in terms of general overall improvements and thus shorter job completion time. As different applications and sometimes even different input files of the same application can have divergent requirements, a set of *different (system) categories* is required, for example providing lots of memory, access to fast locally attached temporary scratch space or fast interconnects.

A requirement for different options is also true for the way underlying resources are provided. As the use of hypervisor-based virtualization can still lead to severe performance degradation [Younge and Fox, 2014] there are several providers offering access to bare-metal clouds, as mentioned before. As an alternative, we propose the use of *Container-Based Virtualization for HPC*. As own research [Gantikow et al., 2015] and general performance evaluation [Felter et al., 2014] has shown, it offers almost bare-metal performance and other advantages, for instance encapsulating dependencies. As security may be considered weaker, compared to hypervisor-based virtualization [Jackson, 2015], this should be especially

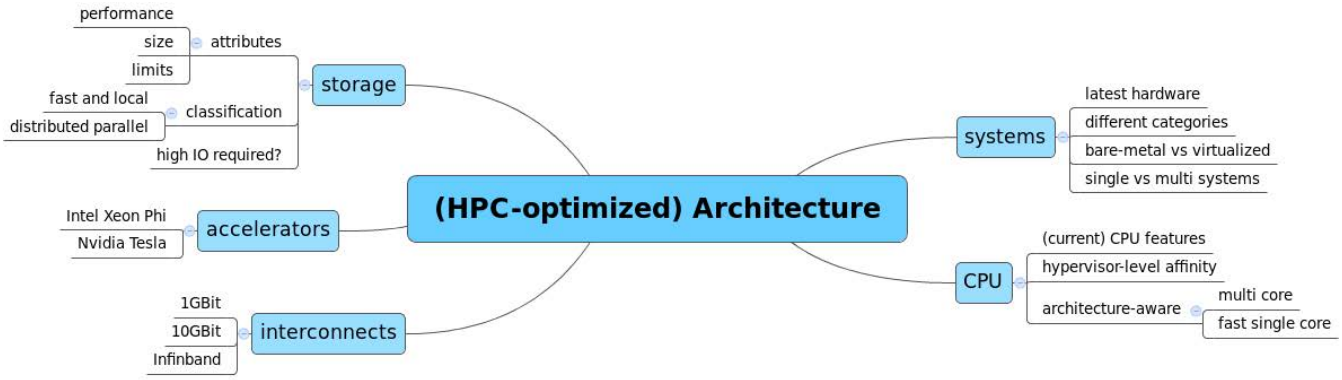


Fig. 4. HPC-aware Clouds Taxonomy Detail: (HPC-optimized) Architecture

considered by private cloud providers and providers offering exclusive access to physical systems, as there is in general less risk from multi-tenancy involved.

Furthermore there's a need for choice when it comes to *single versus multi systems* as some applications perform best with one fast single system, whereas other profit from utilizing multiple systems at once for distributed computations.

HPC codes strongly benefit from latest systems, especially due to CPU micro-architecture advancements. For example the introduction of the Intel Advanced Vector Extensions (AVX) Instructions essentially doubled the floating point capability of Intel processor, which led to significant performance improvements for HPC [Saini et al., 2014]. This shows that a provider has to be aware of improvements and make them available to his customers early on.

For optimizing the performance of HPC applications it is important to pay close attention to details, especially in virtualized environments, for example by utilizing features like *hypervisor-level affinity*. This improves performance [Gupta et al., 2013] when using multiple virtual cores, as it avoids the possible mapping of multiple virtual cores onto the same physical core. Cloud providers should have a deep understanding of performance optimization, as for instance this feature, as opposed to CPU pinning, which binds a single process to a specific (virtual) core, cannot be user-enabled.

As not all HPC codes are yet optimized for systems with many cores, but instead still benefit from fewer CPU(-core)s with higher clock speed, cloud consumers should be able to utilize systems with different characteristic, enabling some *architecture-aware* selection of resources. In case a provider is offering HPCaaS with a service model providing higher abstraction than mere infrastructure components, he should have a deep insight into application and workload requirements for providing reasonable assumptions on the optimal underlying system configuration, as optimal performance usually depends on both an optimized hardware and software stack.

For high throughput and especially low latency for communication middleware such as Message Passing Interface (MPI), which is needed for distributed computations, HPC usually relies on fast *interconnects* like InfiniBand. For certain use cases using 10GBit Ethernet or only 1GBit Ethernet as interconnects may be sufficient, especially if there is very little communication during a distributed computation, but in general Infiniband is favored. As a result close attention should be paid to the placement strategies of requested resources, as low latency and multi-region resources are in contradiction.

Also of interest are *accelerators* like Intel Xeon Phi and NVIDIA Tesla: they are not necessarily essential for all HPC applications, but are getting more and more common in more recent installations and thus getting expected by consumers.

Those specialized resources only became available in Cloud environments recently due to improvements in hardware virtualization features, such as Single Root I/O Virtualization (SR-IOV), finally enabling the use of InfiniBand in virtualized environments [Jose et al., 2013] and bridging the gap between virtualized and bare-metal performance, if tuned right [Musleh et al., 2014].

The availability of these HPC-enablers is currently mostly limited to bare-metal HPC cloud providers, as full support in cloud computing platforms like OpenStack or OpenNebula is still lacking. But when trying to raise the HPC-awareness of an infrastructure one should take this specialized hardware into consideration, as support for virtualized environments is getting better over time.

Another important aspect for certain HPC applications is the availability of access to high-end *storage* systems, as applications can generate several terabyte of scratch or result data. This data can usually be transferred to slower drives after the actual computation, but fast storage is at least required temporary. In general quickly gaining access to vast amounts of storage in clouds is no problem. Businesses like Dropbox for example rely on Amazon Simple Storage Service (S3) - but HPC requires much faster storage backends. In traditional on-site scenarios HPC-suitable storage is commonly provided in the form of fast storage systems, directly attached to a compute node or by utilizing fast distributed parallel filesystems like Lustre [Zhao et al., 2010], GPFS or Ceph [Weil, 2007], again accessed by fast interconnects. As from a cloud consumers perspective the possibility to use fast storage backends makes sense, a HPC-aware cloud provider should be able to offer

these. Both ways of implementation are possible, whereas establishing a performance optimized parallel filesystem shared among multiple customers looks more reasonable from a economic point of view.

D. Service Level Agreements

Service Level Agreements (SLAs) are of major importance for all sorts of cloud services. Going further than regular SLAs and Key Performance Indicators (KPIs) applicable for generic cloud scenarios [Frey et al., 2013], *response time* for instance, HPCaaS has additional requirements, which need to be negotiated in equivalent SLAs.

Figure 5 extends the *Service levels for high performance computing providers* presented in [Kübert, 2014] by an optional *Job runs to completion* SLA, further increasing the suitability of the HPC SLAs for long running jobs in clouds.

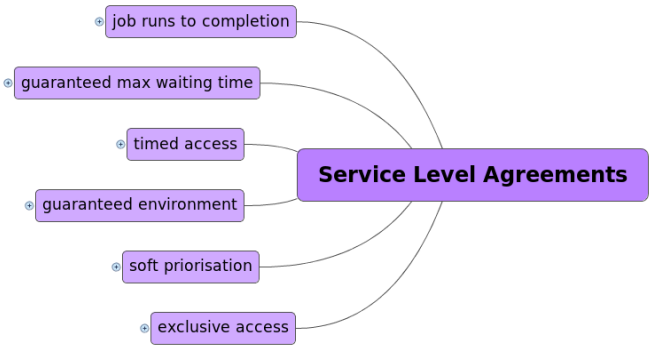


Fig. 5. HPC-aware Clouds Taxonomy Detail: SLAs

This is essential for computations running several days, ensuring a successful termination in cases where the application offers no checkpointing feature, which is also discussed in section II-E. As HPC jobs are in general run on a *best effort* base, meaning they get queued after submitting and then started as soon as sufficient resources become available, this can result in long delays.

In cases where the *job runs to completion* SLA is combinable with a short *guaranteed maximum waiting time* SLA this can result in a significant improvement for customers considering an alternative to local clusters with long waiting time, leading to more QoS-oriented computations.

E. Stability (Subcategory of Security)

Convincing customers planning to process confidential data of the security of a cloud offering can be challenging. Even if services offer all required and applicable certificates and utilizing cloud resources could even provide increased security over the status quo of their on-site environment for some customers, security is still an obstacle for adoption of clouds in general. Transport layers can be secured by accepted standards, data stored in a cloud can be encrypted, but by the time it is getting processed it has to be unencrypted. Even if there has been some research on *homomorphic encryption* [Gentry, 2009], this concept is currently far from suitable for any practical use in the field of HPC.

Our approach is to take the quality of the offered service into consideration, as we think the *stability* is an appropriate instrument to build up trust between cloud provider and consumer if all relevant security standards are implemented.

Figure 6 shows the relevant aspects to consider for the stability of a HPC-aware cloud.

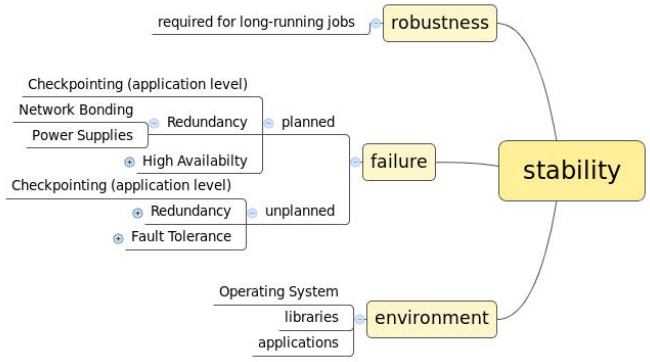


Fig. 6. HPC-aware Clouds Taxonomy Detail: Stability (Subcategory of *Security*)

As there can be long running HPC computations, a provider has to be reduce risks for system *failure*. While for other use cases, an application hosted on a webserver farm running in the cloud for instance, where mechanisms to ensure the availability can be implemented in form of a load balancing proxy, such measures, providing application-level failure protection, do not exist for HPC applications in general. If one node participating in a distributed computation becomes unresponsive this commonly results in an aborted job. Due to this the overall *robustness* of the environment has to assured.

To handle *planned* service interruptions, like downtimes for maintenance work, there are several technical measures widely applied in data centres, for example redundancy of power supplies or network links. In environments using hypervisor-level virtualization High Availability (HA) features like *Live Migration* can be used for moving virtual machines (VMs) from one host to another without service interruption. Such a feature is currently lacking for container-based virtualization, but might change, as *Project CRUI* [CRIU-Project, 2015] is working on implementing checkpoint and restore functionality for Docker containers.

Preparing for *unplanned* interruptions is more complex, as HA features are not sufficient and *Fault Tolerance* is necessary. For virtualized environments there is an option to run VMs in *lockstep* on several hosts. As this requires additional hosts and in some cases has problems with multiple cores, this is in general not to be considered suitable for HPC.

Another aspect of stability is more related to the stability of the computation environment. Some customers need to ensure the repeatability and reproducibility of their computations, for example car manufactures due to product liability. This means that a job has to be re-computable using the exact same

application and library versions as the original computation, available on all participating resources. Providing virtualized resources to customers with such strict requirements can clearly represent an advantage, as it is easier to preserve defined release levels.

III. LEVEL OF REQUIRED ADJUSTMENT

To supply an overview of the *level of required adjustment* needed to offer and utilize HPC resources in clouds we provide table I. It lists an estimate for each of the categories of our taxonomy and rates them with one of the adjustment levels *minor*, *medium* and *major*, required to make a cloud offering suitable for HPC workloads, for both providers and customers.

Our expected level can only represent a rough estimate, as for providers the level of adjustment strongly depends on factors like the desired degree of HPC-awareness and for consumers mainly on the available service model.

TABLE I
ESTIMATED REQUIRED LEVEL OF ADJUSTMENT FOR HPC-AWARENESS OF A CLOUD (FOR BOTH PROVIDER AND CONSUMER)

	Level of Adjustment	
Category	***Provider***	***Consumer***
Lifecycle	minor	major
Security	medium	medium
Business	major	minor
(HPC-optimized) Architecture	major	minor
Operations Management	minor	minor
Service Level Agreements	medium	minor

As the table shows, most of the major adjustments are required for cloud providers, as raising HPC-awareness of their offering requires invest in several supplemental add-ons, optimizations of overall architecture and a deep understanding of HPC requirements. This might be the reason for the limited amount of HPCaaS offerings, as the the number of customers interested in those highly specialized services is only a small fraction of the overall Cloud Computing market.

On the other hand for customers an increased availability of HPC services in clouds would be quite worthwhile, as basically all their major adjustments need to focus on their workflows related to computing. If these are adaptable with justifiable effort, HPCaaS can considerably reduce their investment costs related to HPC systems and reduce turnaround times of compute jobs.

IV. CONCLUSION

Our taxonomy shows that there are many aspects to consider if cloud resources have to be used for HPC, even though the concept itself seems self-evident, especially for SMEs.

HPC has a need for lots of resources and providing them in a suitable performance-optimized way requires adjustment on several general aspects and even more on details. Performance wise the use of virtualization is getting less of an issue, especially with the support of the discussed HPC enablers and operating system level virtualization providing near bare-metal performance and also improving the packaging, delivering and access to complex software.

From a workflow perspective a transition to cloud services always results in a modification of processes, which is true for HPC too. For HPC this is especially tasking due to data storage and transfers of often huge amounts of data, but can be eased by hosting the data directly at the provider.

But the hardest aspect to solve is the element of *trust*. Even in cases when using cloud services can be considered secure enough this can be an aspect limiting their adoption. This is why we consider *future research* on improving workflow automation aspects with regard to security as important, as not all jobs require guaranteed privacy by running in a private data center, but could be processed at a cloud provider.

REFERENCES

[Bull, 2015] Bull (2015). Bull Extreme Factory. http://www.bull.com/extreme-factory.

[CRIU-Project, 2015] CRIU-Project (2015). Checkpoint/Restore In Userspace (CRIU). http://www.criu.org/.

[Felter et al., 2014] Felter, W., Ferreira, A., Rajamony, R., and Rubio, J. (2014). An updated performance comparison of virtual machines and linux containers. *technology*, page 28:32.

[Frey et al., 2013] Frey, S., Reich, C., and Lüthje, C. (2013). Key performance indicators for cloud computing slas. In *The Fifth International Conference on Emerging Network Intelligence, EMERGING*, pages 60–64.

[Gantikow et al., 2015] Gantikow, H., Klingberg, S., and Reich, C. (2015). Container-Based Virtualization for HPC. In *Proceedings of CLOSER 2015*.

[Gentry, 2009] Gentry, C. (2009). *A fully homomorphic encryption scheme*. PhD thesis, Stanford University. crypto.stanford.edu/craig.

[Gupta et al., 2013] Gupta, A., Kal, L. V., Gioachin, F., March, V., Suen, C. H., Lee, B.-S., Faraboschi, P., Kaufmann, R., and Milojicic, D. S. (2013). The who, what, why, and how of high performance computing in the cloud. In *CloudCom (1)*, pages 306–314. IEEE Computer Society.

[Jackson, 2015] Jackson, I. (2015). Surviving the Zombie Apocalypse – Security in the Cloud Containers, KVM and Xen. http://xenbits.xen.org/people/iwj/2015/fosdem-security/.

[Jose et al., 2013] Jose, J., Li, M., Lu, X., Kandalla, K., Arnold, M., and Panda, D. (2013). SR-IOV Support for Virtualization on InfiniBand Clusters: Early Experience. In *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, pages 385–392.

[Kübert, 2014] Kübert, R. (2014). *Service Level Agreements for Job Submission and Scheduling in High Performance Computing*. dissertation, Universität Stuttgart.

[Lab, 2015] Lab, C. R. (2015). High performance computing in the cloud. http://www.wolke.hs-furtwangen.de/currentprojects/hpc-in-the-cloud.

[Musleh et al., 2014] Musleh, M., Pai, V., Walters, J., Younge, A., and Crago, S. (2014). Bridging the Virtualization Performance Gap for HPC using SR-IOV for InfiniBand. In *Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on*, pages 627–635.

[Penguin Computing, 2015] Penguin Computing (2015). Penguin Computing On-Demand: POD. https://pod.penguincomputing.com/.

[Saini et al., 2014] Saini, S., Chang, J., and Jin, H. (2014). Performance evaluation of the intel sandy bridge based nasa pleiades using scientific and engineering applications. In Jarvis, S. A., Wright, S. A., and Hammond, S. D., editors, *High Performance Computing Systems. Performance Modeling, Benchmarking and Simulation*, volume 8551 of *Lecture Notes in Computer Science*, pages 25–51. Springer International Publishing.

[UberCloud, 2015] UberCloud (2015). The UberCloud Experiment. https://www.theubercloud.com/hpc-experiment/.

[Weil, 2007] Weil, S. A. (2007). *Ceph: reliable, scalable, and high-performance distributed storage*. PhD thesis, UNIVERSITY OF CALIFORNIA SANTA CRUZ.

[Younge and Fox, 2014] Younge, A. and Fox, G. (2014). Advanced Virtualization Techniques for High Performance Cloud Cyberinfrastructure. In *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*, pages 583–586.

[Zhao et al., 2010] Zhao, T., March, V., Dong, S., and See, S. (2010). Evaluation of a performance model of lustre file system. In *ChinaGrid Conference (ChinaGrid), 2010 Fifth Annual*, pages 191–196.

Event detection using adaptive thresholds for non-intrusive load monitoring

Frederik Laasch
Université de Haute Alsace
Mulhouse, France
Email: laaschfr@hs-furtwangen.de

Alain Dieterlen
Université de Haute Alsace
Mulhouse, France
Email: alain.dieterlen@uha.fr

Dirk Benyoucef
Furtwangen University
Furtwangen, Germany
Email: dirk.benyoucef@hs-furtwangen.de

***Abstract*—This paper deals with event detection applied to non-intrusive load monitoring. In the course of the paper a scheme to improve the event detection for non-intrusive load monitoring is introduced. It is shown, that by adding adaptive thresholds to an event detector it is possible to decrease the amount of missed or falsely detected events. As reference two event detectors are used. Their performance is evaluated and the causes for detecting non-existing events and missing events are pointed out. Finally, it is shown that the performance of the aforementioned event detectors is improved by applying time dependent thresholds.**

I. Introduction

Today, one of the most pressing topics is energy consumption and its reduction. Monitoring the energy consumption is done by using a smart meter and Non-Intrusive Load Monitoring (NILM). Using NILM makes it possible to determine the power consumption of each appliance, using only one smart meter, instead of equipping each appliance with its own measuring unit. The detailed information gained by applying NILM is used to analyze appliance behavior and optimizing their usage in order to achieve energy savings.

Based on the measurements, the event detection of a NILM system determines the point in a time dependent signal where a transient occurs. Between two transients there is a steady-state. In a NILM system the time of occurrence of a transient and the duration of a steady-state are of interest. Being aware of the times events occur, it is possible to perform a classification. The classification determines the probable composition of the aggregated signal. Knowing the energy consumption and its components, it is possible to perform energy tracking to determine the energy consumption of each appliance.
This paper deals with event detection. A reliable event detection is crucial for the classification and the energy tracking. Assuming that event detection fails, classification and energy tracking are not working well. The cause for event detection failure is highlighted and a method to improve the event detection is introduced.

Section II reviews the state of the art. In section III the dataset used in this paper is introduced. Section IV presents the event detectors considered and section V shows the problems inherent to the event detectors. Sections VI and VII introduce the developed solution and its performance evaluation.

II. State of the art

In this paper the focus is on event detectors applying a threshold to a signal. Furthermore, the focus is limited to event detectors that are used in NILM. The following summary of event detectors is by no means exhaustive but provides an overview about techniques used for event detection in NILM.
In [1], [2] a threshold is applied to the mean of the power signal. In [3], the differentiation of the power signal is thresholded. In [4], the differentiation of the power signal is calculated. If the change in the differentiated signal is bigger than $10\,\%$ an event is detected. In [5], the harmonics of the power signal are searched for patterns indicating the occurrence of an event. The patterns are derived by calculating the mean of the harmonics, and searched for relevant changes during a training phase. In [6], the mean of the power signal is windowed by two adjacent windows and the frequency distributions of the windows are compared. When the difference of the frequency distributions is above a threshold an event is triggered.

III. The dataset

For the simulations done in this paper measurements of various appliances are used. The measurement data provides ground truth information, i.e. each event is marked. The data was collected during two different campaigns. In the first campaign single appliances were recorded. Altogether 2474 events were recorded. The dataset is called "D1". The appliances, which were recorded without simultaneous operation were: two fridges, an immersion heater, an iron, a kettle, a mixer and three ovens. In the second campaign multiple appliances were hooked up to the grid at the same time and 2893 events were recorded. The second dataset is called "D2". Datasets D1 and D2 feature sampling rates of 1 kHz. Both campaigns were performed at Furtwangen

University.
In [7], the measurement system which is used to collect the ground truth data is described. The system incorporates a switching box, which toggles the appliances and therefore enables the detection of events. The event markers have an error of approximately 100 ms.

IV. Algorithm description

Two event detectors are used. The two event detectors are the ones introduced by Hart (H1) [3] and by Berges (B1) [6]. The performance of the event detectors suffers from multiple events caused by an appliance's single switching process. In the following the event detectors are described and the choice of parameters is explained. The parameters are chosen to enable best performance using a fixed threshold. The results achieved based on those parameter sets are used as reference later on.

The event detector H1 applies a threshold to the differentiated power signal. When the differentiated power signal is above the threshold, an event is detected. Hart normalized the power signal to account for fluctuations in the grid voltage.
Instead of the change of the signal, as exploited by H1, the event detector B1 uses the probability distribution of the power signal. Two adjacent windows are applied to the signal. The probability distributions of the windows are estimated and compared using a χ^2-test. When the test finds the distributions unequal, an event is detected. The event detector B1 is applied to the mean of the power signal calculated over one period of the power signal.

The parameters selected for the event detectors are geared to the needs of the data set which were introduced in sec. III. The threshold for the approach H1 was chosen so that events of the appliances with the smallest change in the mean of the power signal are still detectable – but high enough so that the threshold is above the noise level. The threshold is set to 10 W.
The approach B1 had a window length of 20 datapoints, 80 bins and a threshold of 80 W. The length of the windows was selected to be shorter than the length of the smallest event in the data pool. The number of bins enables the detection of the event with the least change. The threshold was selected applying the same reasoning as with Hart's approach.

V. Problem description

In this section, it is shown that one device caused multiple events, when switched ON or OFF. The causes for those multiple events will be analyzed and a strategy for canceling those falsely detected events is introduced.

All the event detectors introduced are applied to the mean of the instantaneous power. The mean is used in order to remove the periodicity of the signal. Applying the same preconditioning for all event detectors enables comparison between them. $\overline{P}$ is the mean of the instantaneous power. N is a period of the instantaneous power. $\overline{P}$ is calculated as follows:

$$\overline{P}(k) = \frac{1}{N} \sum_{n=0}^{N-1} P(k+n) \tag{1}$$

$\overline{P}$ contains peaks, noise and damped or undamped oscillations. Often the events are followed by over- or undershoots. Those disturbances of $\overline{P}$ are caused by the composition of the appliance. Since the property of $\overline{P}$ exploited by the event detector is affected by those disturbances the quality of the event detection is affected too; the distortions are detected as additional events.

In the case of Hart's approach, each change in the signal and especially fast changes cause corresponding changes in the differentiated signal. In Fig. 1 $\overline{P}$ and $\frac{d}{dk}\overline{P}$ of a freezer are shown. The figure shows one turn ON and one turn OFF event. However applying the threshold, shown on the graph multiple events are detected. When using an event detector that relies on the differentiation of a signal, there are problems with suppressing those additional events.
Because of the offset of the event marker (sec. III), multiple events within 100 ms after the turn on event cannot be distinguished. For example, the freezer's rising slope in $\overline{P}$ is approximately 120 ms in length. Therefore only the events within the ellipse in Fig. 1 can be identified as false events without a doubt. The events in the rectangles are approached as followed: the first event detected is considered to be the true event and the following events are considered to be false events. The multiple events caused by a turn OFF event have to be approached in the same way since the event marker of the turn OFF event is facing the same problem as the one of the turn ON event.

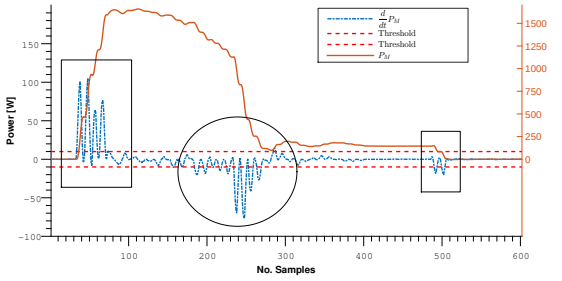


Fig. 1: Section of $\overline{P}$ and its differentiation

Berges compares two frequency distributions using a χ^2-test. If the difference of the distributions defined in Eq. (2) is higher than a threshold, an event is triggered. In Eq. (2) N is the number of bins. y_i and

x_i are the numbers of samples in bin i of the first and second window. Disturbances have an impact on the distributions.

$$\chi^2 = \sum_{i=1}^{N} \frac{(y_i - x_i)^2}{x_i} \tag{2}$$

At a certain point each of the above mentioned event detectors apply a threshold to a property of $\overline{P}$. There are disturbances that cause the property of $\overline{P}$ to rise above the threshold.

One possibility is to blank out all events after an event occurred for a period of time. But blanking out those events might lead to missing events caused by other appliances.

A better approach is to adapt the threshold applied to the property of $\overline{P}$. For the event detector H1 this corresponds to adapting the change over time in $\overline{P}$ triggering an event. Considering the approach B1, the adaptation of the threshold results in varying allowed divergences between the two monitored probability distributions. In the following section the adaptation of the thresholds is further invested.

VI. Approach

The cancellation of the events caused by the disturbances in the monitored property of $\overline{P}$ is done by adapting the applied threshold. Again, $\overline{P}$ is the mean of the instantaneous power and the mean is applied in order to remove the periodicities of the power. Adding an adaptive threshold to H1, the resulting event detector is called H2. Likewise the combination of B1 and an adaptive threshold is called B2. The adaptation of the threshold in H2 is shown in Fig. 2. The box $\frac{d}{dk}$ is the differentiator applied H2. For adapting the threshold in B2 the box $\frac{d}{dk}$ in Fig. 2 is replaced by the calculation of frequency distributions over windows. *Threshold generation* is the adaptation of the threshold and *Event detection* applies the adapted threshold to the signal.

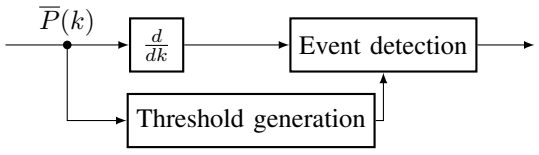


Fig. 2: Approach for adapting a threshold

The adaptation of the threshold in H2 is done by using the standard deviation of $\overline{P}$. For the generation of a time dependent threshold $T(k)$ a preselected factor C is multiplied with the mean of the standard deviation $\overline{\sigma}_{\overline{P}_M}$ of the mean $\overline{P}_M$ of $\overline{P}$. C is a constant that accounts for measurement noise. Before multiplying by C it is checked if $\overline{\sigma}_{\overline{P}_M}(k)$ is smaller than one. In the case that $\overline{\sigma}_{\overline{P}_M}(k)$ is smaller than one, $\overline{\sigma}_{\overline{P}_M}(k)$ is set to one.

$$\overline{P}_M(k) = \frac{1}{L} \sum_{l=0}^{L-1} \overline{P}(k-l) \tag{3}$$

$$\sigma_{\overline{P}_M}(k) = \begin{cases} 1, & \sigma_{\overline{P}_M}(k) < 1 \\ \sqrt{\left(\overline{P}(k) - \overline{P}_M(k)\right)^2}, & \text{else} \end{cases} \tag{4}$$

$$\overline{\sigma}_{\overline{P}_M}(k) = \frac{1}{M} \sum_{m=0}^{M-1} \sigma_{\overline{P}_M}(k-m) \tag{5}$$

$$T(k) = C \cdot \overline{\sigma}_{\overline{P}_M}(k) \tag{6}$$

The time dependent threshold applied in B2 is calculated by applying Eq. (3) to Eq. (5) and by replacing Eq. (6) by Eq. (7). Instead of multiplying C by the standard deviation the standard deviation is added to C. The observation window used to calculate the adaptive threshold is defined by Eq. (3).

$$T(k) = C + \overline{\sigma}_{\overline{P}_M}(k) \tag{7}$$

VII. Simulations

In order to verify the selected approach the data described in sec. III is used for simulations. First the event detectors H1 and B1 are tested. Second the adapted versions of the event detectors are used for simulations (H2, H3, B2). The adaptation is done in accordance with the approaches introduced in section VI.

In order to evaluate the performance of the different event detectors the metrics True Positive Rate (TPR) and False Positive Rate (FPR) [8] are applied. The TPR and FPR are defined as follow:

$$TPR = \frac{TP}{TP + FN} \in [0, 1] \tag{8}$$

$$FPR = \frac{FP}{FP + TN} \in [0, 1] \tag{9}$$

In addition to those metrics the power difference between the threshold and the power at the time instances where false positives or false negatives occurred is calculated. Those power differences are measures used to evaluate the goodness of the applied adaptive threshold, for example a false negative with a small power difference is less grave than the ones with a large power difference. In Eq. (10) and Eq. (11) P_{FN} and P_{FP} are those power differences and FN_{idx} and FP_{idx} are vectors containing the indexes of the false or missed events respectively. A is the total number of false negatives and B is the total number of false positives.

$$P_{FN} = \sum_{a=1}^{A} \left[\overline{P}(FN_{idx}(a)) - T(FN_{idx}(a))\right]^2 \tag{10}$$

$$P_{FP} = \sum_{b=1}^{B} \left[\overline{P}(FP_{idx}(b)) - T(FP_{idx}(b))\right]^2 \tag{11}$$

	TPR	FPR $\cdot 10^{-6}$	$P_{FP} \cdot 10^9$	$P_{FN} \cdot 10^9$
B1	0.7566	142.5	57.397	0.936
H1	0.7946	59.89	6.952	0.611
B2	0.5691	67,96	53.669	32.019
H2	0.6200	7.832	0.474	0.876
H3	0.7821	15.18	1.211	0.731

Table I: Simulation results based on event detectors with a fixed and a time depending threshold. Dataset D1 was used. There are 2474 events in dataset D1.

	TPR	FPR $\cdot 10^{-6}$	$P_{FP} \cdot 10^9$	$P_{FN} \cdot 10^9$
B1	0.4998	$9.18 \cdot 10^3$	2.929	3.6066
H1	0.6097	127.8	88.789	2.921
B2	0.0839	1.890	0.461	5.823
H2	0.3802	$896.6 \cdot 10^{-3}$	0.314	5.077
H3	0.6505	4.733	2.322	2.759

Table II: Simulation results based on event detectors with a fixed and a time depending threshold. Dataset D2 was used. There are 2893 events in dataset D2.

In Table I the results for the simulations using dataset D1 are presented. Table II shows the results achieved with dataset D2. The upper parts of the tables show the simulation results for the unmodified event detectors (B1, H1) and the lower parts show the results for the event detectors applying an adaptive threshold (B2, H2, H3).

The parameters used for B1 and H1 are the ones introduced in sec. IV. B2 and H2 are using the same parameters. Additionally the parameters for the adaptation of the threshold are chosen as followed: $L = 5$, $M = 20$ and C is the same value as the value for the fixed threshold. H3 is using the same parameters but C is set to 5.
The choice of parameters was done in order to minimize the error functions in Eq. (8) to Eq. (11).

For dataset D1 H2 resulted in the lowest FPR. Compared to H1, the FPR was reduced by $52.058 \cdot 10^{-6}$. The drawback is that the TPR has decreased by 0.1746.
H3 has a FPR which is lower than the FPR of H1 by $10.447 \cdot 10^{-6}$. But the TPR only decreased by 0.0125 compared to H1. Looking at P_{FP} and P_{FN} the result is the same.
B2 achieved a reduction of the FPR by $74.54 \cdot 10^{-6}$ but at the same time the TPR decreased by 0.1875 compared to B1.

Regarding dataset D2, the approach using H3 achieved the highest TPR. The FPR decreased by $123.067 \cdot 10^{-6}$ and the TPR increased by 0.0408. H2 reduced the FPR by $126.9034 \cdot 10^{-6}$. But in exchange the TPR decreased by 0.2295. Considering P_{FP} and P_{FN} the outcome is the same. If the task is to reduce the FPR as much as possible one must use H2. For the best TPR H3 must be used.
The approach based on B2 reduced the FPR by $9.178 \cdot 10^{-3}$ in comparison to B1. In the same time the TPR decreased by 0.4159 compared to B1.

VIII. Conclusion

In this paper a novel approach to event detection in non-intrusive load monitoring was introduced. Applying adaptive thresholds to the event detection, the number of false positives caused by a single event has been reduced. The event detectors of Hart [3] and Berges [6] were extended by adaptive thresholds. The results achieved with the modified event detectors are compared to the results achieved with the original ones. The simulations done to verify the new approach were done based on datasets collected at Furtwangen University.

With respect to the approaches of Hart and Berges, the overall performance of the event detection is improved. In order to achieve those results, different sets of parameters were tested. Evaluating the tests, necessary steps for choosing the parameters of the event detectors became obvious. First the parameters of the unmodified algorithms have to be optimized. Second, the parameters for the adaptation of the threshold have to be chosen.

Further work will be focused on the verification of the results. The plan is to apply the event detectors with time dependent thresholds to different public datasets. Since the adaptation of the event detectors seems to be very promising, further algorithms will be adapted for usage with time dependent thresholds.

References

[1] D. Egarter and W. Elmenreich, "Autonomous load disaggregation approach based on active power measurements," http://128.84.21.199/pdf/1412.2877.pdf, 2014.

[2] D. C. Bergman, J. Dong, J. P. Juen, N. Tanaka, C. A. Gunter, and A. K. Wright, "Distributed non-intrusive load monitoring," in *2011 IEEE PES Innovative Smart Grid Technologies (ISGT)*. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5759180: IEEE, January 2011, pp. 1–8.

[3] G. Hart, "Non-intrusive Appliance Load Monitoring," in *JRC Technical Report*. Proceedings of the IEEE: IEEE, December 1992, pp. 1870–1891.

[4] A. S. Ardeleanu and C. Donciu, "Nonintrusive load detection algorithm based on variations in power consumption," in *2012 International Conference and Exposition Electrical and Power Engineering*. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6463913: IEEE, 2012, pp. 309–313.

[5] S. Leeb, S. Shaw, and J. Kirtley, "Transient event detection in spectral envelope estimates for nonintrusive load monitoring," in *IEEE Transactions on Power Delivery 10*. IEEE, Juli 1995, pp. 1200–1210.

[6] Y. Jin, E. Tebekaemi, M. Berges, and L. Soibelman, "Robust adaptive event detection in non-intrusive load monitoring for energy aware smart facilities," in *Proceedings of the IEEE International Conference on Acoustics, Speech,and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic*, 2011, pp. 4340–4343.

[7] T. Bier, D. Benyoucef, D. O. Abdeslam, J. Merckle, and P. Klein, "Smart meter systems measurements for the verification of the detection and classification algorithms," in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2013, pp. 5000–5005.

[8] K. D. Anderson, M. E. Berges, A. Ocneanu, D. Benitez, and J. Moura, "Event detection for non intrusive load monitoring," in *38th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 2012, pp. 3312–3317.

A Virtual-Reality 3d-Laser-Scan Simulation

Malvin Danhof, Tarek Schneider, Pascal Laube, Georg Umlauf

Institute for Optical Systems, University of Applied Sciences Constance, Germany

***Abstract*—We present a 3d-laser-scan simulation in virtual reality for creating synthetic scans of CAD models. Consisting of the virtual reality head-mounted display Oculus Rift and the motion controller Razer Hydra our system can be used like common hand-held 3d laser scanners. It supports scanning of triangular meshes as well as b-spline tensor product surfaces based on high performance ray-casting algorithms. While point clouds of known scanning simulations are missing the man-made structure, our approach overcomes this problem by imitating real scanning scenarios. Calculation speed, interactivity and the resulting realistic point clouds are the benefits of this system.**

I. Introduction

In science and industry 3d laser scanners are widely used to acquire 3d point data of real world objects. The result of a scanning process is a 3d point cloud. Often a CAD representation of these point clouds needs to be recovered for the subsequent processing. This task is performed by the reverse engineering process. Thus, the quality of the CAD representation depends on the chosen reverse engineering process.

Evaluating reverse engineering algorithms is only possible if a large set of point clouds is available. To acquire these points clouds CAD models are scanned virtually. There are two reasons for this approach. First, we often lack enough suitable physical objects to scan and, second, point clouds of hand-scanned physical objects often lack corresponding CAD information. However, point clouds of hand-scanned physical objects and synthetically generated point clouds differ heavily in their structure in terms of scan-strategy and scan-path. Since each human operator has a different scan-strategy and scan-path, the resulting point clouds differ much in structure, even if the same object was scanned. On the one hand this structure is not completely random, on the other hand there is no good model for the human scanning procedure. For a fair evaluation of reverse engineering algorithms with realistic data this man-made structure must be incorporated into the data. To generate scans of CAD models that capture this man-made structure we propose a virtual reality (VR) scanner setup. We present a method to generate 3d point clouds from CAD models consisting of triangle meshes and b-spline tensor product surfaces with a simulated hand-held laser scanner in a VR environment. Our goal is to create a realistic simulation of the scanning process with a hand-held laser scanner like the FARO Edge ScanArm (Fig. 1). Using this approach we can compute 3d point clouds of CAD models using a virtual laser scanner with a man-made scan structure.

Fig. 1: FARO Edge ScanArm (www.faro.com, 09/25/2015)

This paper is laid out as follows: First we describe our experimental setup and the used peripherals in Chapter II. Chapter III outlines parameters of the scanning simulation and the used data structures. We then explain the process of ray-casting for triangle meshes in Chapter IV and for b-spline surfaces in Chapter V. Results are presented in Chapter VI.

A. Related Work

The two most commonly used methods for high accuracy 3d scanning are laser scanners and structured light scanners. While hand-held laser scanners project a laser line onto the object surface structured light laser scanners project a pattern of light. Especially since the Michelangelo Project [1] laser scanning systems have received increasing attention. Resulting point clouds are used in numerous research fields. Ip and Gupta [2] are retrieving matching CAD models by using partial 3d point clouds. They use real and synthetic point clouds. To generate the synthetic point clouds CAD surfaces are evaluated at random parameters. Mitra et al. [3] register two point clouds by minimization of the squared distance between the underlying surfaces. They use synthetic point clouds from random evaluation with an unspecified noise. Bernardini and Bajaj [4] use synthetic point clouds generated by sampling a surface uniformly for an automatic reconstruction process. In [5] Tagliasacchi et al. extract curve skeletons from incomplete point clouds. They use a bounding sphere around a CAD model at its center to get different viewpoints. These viewpoints are used for orthographic ray casting from a uniform grid. A very similar approach is used in [6] where a fan of rays with origin on a bounding sphere is swept along the model surface from different viewpoints. However, all these approaches do not capture a realistic scan structure.

The application of VR is widespread in various scientific areas. Especially in medical research VR has a huge potential using it for surgery trainings [7] or the medication in occupational therapy [8]. In the future, there will be many different applications of virtual or augmented reality simulations. A collaborative approach to develop those applications is given by [9] which also contains a collection of motion control techniques. Numerous different tools are available and individual techniques should be selected according to a concrete target application at hand.

II. Experimental setup

For the simulation of a hand-held 3d laser scanner we propose a VR environment. The structure of the synthetic point clouds generated in this VR is similar to man-made scans of physical objects due to the manual scanning process (Figure 2). The virtual scene consists of a workshop with a table, that carries the CAD model, and a virtual model of the scanner, see Figure 3.

Fig. 2: VR setup.

The user can move freely in the scene, change the perspective and field of view in terms of the CAD model and operate the laser scanner.

For the VR setup two essential peripherals are used (Figure 4): the VR headset Oculus Rift [10] for visual immersion and the motion controller Razer Hydra [11] for scanner control and scene navigation.

The Oculus Rift provides a 3d view of the VR scene and allows to freely explore the 3d scene. Integrated sensors provide data about the user's head position and orientation. Due to its large field of view the degree of immersion in the scene is very high.

The Razer Hydra consists of two wired controllers and a base station, which creates a magnetic field to determine the controllers' spatial positions. It can capture hand movements and orientations accurately and provides joysticks and buttons for control tasks.

Both devices are integrated using the manufacturer' APIs: Oculus Rift SDK [10] and Sixense Core API [11].

III. Simulation parameters and data structure

A. Laser line probe

The simulation is based on ray-casting computing intersections of a ray and a triangle mesh or b-spline tensor product surface. The ray is emitted by a virtual laser line probe. The virtual scanner simulates a real laser scanner by sweeping a planar cone of $n+1$ rays $R_0, \ldots, R_n$ over the surface, see Figure 5. The rays originate from scanner position O and lie

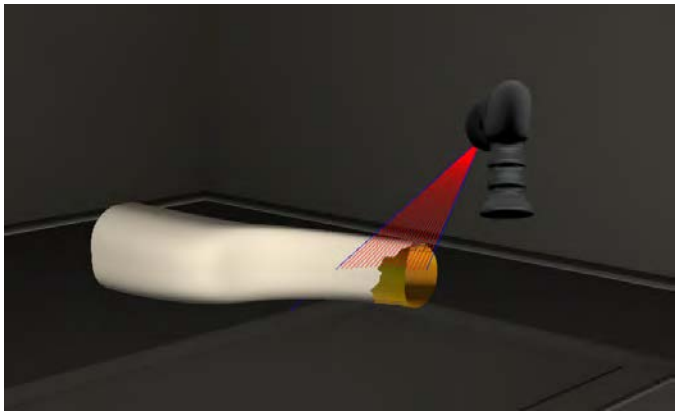


Fig. 3: The virtual scene.

Fig. 4: Left: Oculus Rift (www.giga.de, 09/25/2015), Right: Razer Hydra (www.roadtovr.com, 09/25/2015)

within a cone with axis D and aperture angle φ. Two rays R_{i-1} and R_i have angle φ/n for $i = 1, \ldots, n$.

The orientation of the operator's hand controls the view direction D. The number of rays n and the aperture angle φ are adjustable. The scan line S is the set of intersection points of the rays $R_0, \ldots, R_n$ with the surface.

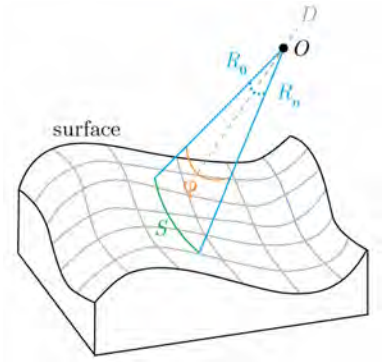


Fig. 5: The setup of the simulation of a laser scanner.

The laser line probe model contains two types of noise. The first noise affects the scanner position O, where a random offset is added. This noise affects all intersection points of one cone equally. The second noise affects the distance of each intersection point from O by modifying the direction of each ray slightly. Both noises are defined by a normally distributed offset with zero mean and user defined variance.

B. Data structure

An octree with axis aligned bounding boxes (AABBs) as nodes is used to partition the bounding box of the CAD model recursively. For a triangle mesh, the triangles are directly inserted into the octree using a triangle-box overlap algorithm [12]. For a b-spline surface, the surface is segmented to determine the parametric interval where the ray intersects the surface. Each segment is bounded by its segment bounding box (SBB). The axis aligned SBBs are inserted into the octree's AABBs by checking their corner coordinates. The octree's leaf nodes contain the triangles or SBBs. To eliminate non-intersecting geometry, ray-box-intersections are tested as in [13, p. 741-744]. If the ray-box-intersection test is positive, the octree is recursively traversed to a leaf node. This process yields a set of leaf nodes possibly containing the actual surface intersection. The geometry inside each of these leaf nodes is tested for intersections (see Chapters IV and V). Finally, the intersection point closest to the scanner position O is selected as the correct ray-surface intersection.

IV. Ray-casting for triangle meshes

Intersection testing for an arbitrary ray with a triangle in 3d is one of the most important non-trivial operations in ray-tracing oriented rendering. The presented solution [14] first checks if a given ray R from P_0 to P_1 intersects a plane P spanned by triangle T with normal n. If P is intersected, the intersection point $P(r_I)$ is checked to be inside T.

First, the intersection point $P(r_I)$ of the ray $R = P_0 + r(P_1 - P_0)$, $r \in \mathbb{R}$, with P is computed. $P(r_I)$ has the parameter

$$r_I = \frac{n \cdot (V_0 - P_0)}{n \cdot (P_1 - P_0)} \tag{1}$$

on R. A valid intersection between R and P occurs only if the denominator of (1) is nonzero and r_I is real with $r_I \geq 0$.

Second, the coordinates P_I of $P(r_I)$ in the plane are computed. A parametrisation of P is given by

$$V(s,t) = V_0 + s(\underbrace{V_1 - V_0}_{u}) + t(\underbrace{V_2 - V_0}_{v})$$

where V_1, V_2, V_3 are the corners of T and $s, t \in \mathbb{R}$. $P_I = V(s_I, t_I)$ is inside the triangle T if

$$s_I \geq 0, \quad t_I \geq 0 \quad \text{and} \quad s_I + t_I \leq 1.$$

P_I is on an edge of T if one of the conditions

$$s_I = 0, \quad t_I = 0 \quad \text{or} \quad s_I + t_I = 1$$

is true. Each condition corresponds to one of T's edges, see Figure 6. In order to compute s_I and t_I, we use barycentric coordinate computation using a 3D generalized perp operator on P as in [15]. With $w = P_1 - V_0$, which is a vector in P, we solve the equation

$$w = su + tv.$$

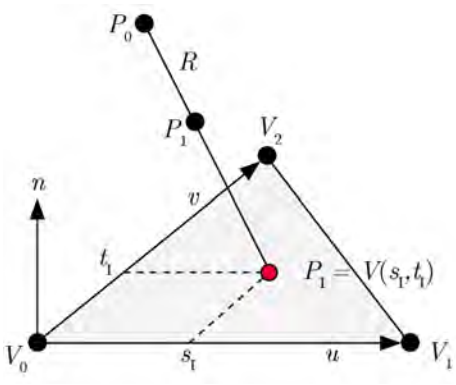


Fig. 6: Ray-triangle intersection

The final result uses five dot products

$$s_I = \frac{(u \cdot v)(w \cdot v) - (v \cdot v)(w \cdot u)}{(u \cdot v)^2 - (u \cdot u)(v \cdot v)},$$
$$t_I = \frac{(u \cdot v)(w \cdot u) - (u \cdot u)(w \cdot v)}{(u \cdot v)^2 - (u \cdot u)(v \cdot v)}.$$

V. Ray-casting for b-spline surfaces

A non-uniform b-spline tensor product surface is defined as

$$s(u,v) = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} N_i^p(u) N_j^q(v).$$

$N_i^p(u)$ and $N_j^q(v)$ are the b-spline basis functions of degree p and q while $c_{i,j}$ are the surface control points. The following ray-casting approach is based on [16]. To find the exact ray-surface intersection point, Newton's method is used. This method requires a good initial guess for parameters u and v of the intersection point. Therefore, we first subdivide the surface into simpler, close to linear, surface segments. These segments are enclosed in SBBs. Testing the ray for intersections with the SBBs, we can eliminate segments that will not be intersected. If a box is intersected, the median of the parametric interval in which the ray possibly intersects the surface, is used as initial guess. Convergence of Newton's method is tested for each intersected SBB.

A. Surface refinement

Each SBB encloses a surfaces segment defined over one knot interval, see Figure 7. In order for Newton's method to converge fast and robustly, it is necessary that the initial guess $(u*, v*)$ is already close to the root. To achieve this, the control point mesh, respectively the knot vector, is refined such that each segment fulfills a curvature-based flatness criterion. The extent of refining a segment defined over knot-span $[t_i, t_{i+1})$ is given by the product of its maximum curvature κ and its arc length δ. Regions with high curvature should be subdivided to avoid multiple roots. Long curve segments should be subdivided to ensure that the initial guess is reasonably close to the root. The heuristic for the number of knots n which will be added to a given knot-span is therefore

$$n = C \cdot \max_{[t_i, t_{i+1})} (\kappa) \cdot \delta^{3/2}, \tag{2}$$

where C allows to control the extent of the refinement. δ is estimated by the sum of distances of sampled segment points. Since a rough guess of maximum curvature is sufficient, a simplified calculation

$$\max_{[t_i,t_{i+1})}(\kappa) \approx \frac{\max_{[t_i,t_{i+1})}(|c''(t)|)}{\text{avg}_{[t_i,t_{i+1})}(|c'(t)|^2)}$$

can be used. This heuristic is applied to each row of the knot grid. The maximum number of knots over all rows determines the number of knots that will be inserted into each u interval. This process is repeated for all columns.

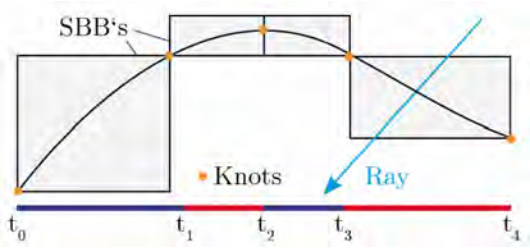


Fig. 7: A b-spline's segments bounded by SBBs in between knot intervals and intersected by ray R.

B. Segment bounding boxes

To acquire tight bounding boxes which completely enclose the part of the surface between two knots, the multiplicity of each knot is increased to the degree of the surface. This yields the Bézier representation of the segments. Using the convex hull property of the Bézier representation coordinate-wise yields axis aligned SBBs, which are inserted into the octree data structure.

C. Root finding

For root finding the ray is described as the intersection of two planes $P_1 = (N_1, d_1)$ and $P_2 = (N_2, d_2)$ in Hessian form. N_1 and N_2 are the orthogonal vectors of unit length, perpendicular to the ray. The roots $(u*, v*)$ of

$$F(u,v) = \begin{pmatrix} N_1 \cdot S(u,v) + d_1 \\ N_2 \cdot S(u,v) + d_2 \end{pmatrix}$$

yield the intersection points. F measures the distance of the evaluated point (u, v) on the surface to both planes. These roots are computed using Newton's method. It converges quadratically if the initial guess is close to the root and the root is simple, which is ensured by the preceding refinement. Newton's iteration for the two parameters u and v is described as

$$\begin{pmatrix} u_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} u_n \\ v_n \end{pmatrix} - J^{-1}(u_n, v_n)F(u_n, v_n), \quad (3)$$

where the Jacobian matrix J of F has the form

$$J = (F_u, F_v) = \begin{pmatrix} N_1 \cdot S_{\partial u}(u,v) & N_1 \cdot S_{\partial v}(u,v) \\ N_2 \cdot S_{\partial u}(u,v) & N_2 \cdot S_{\partial v}(u,v) \end{pmatrix}.$$

Using the initial guess for u and v the iteration is started. This process is iterated until the stopping criterion is met. The iteration is stopped when the difference of the parametric values of successive iterations is smaller than pre-defined ε

$$|u_n - u_{n-1}| + |v_n - v_{n-1}| < \varepsilon.$$

The resulting parameters provide the intersection point P_I.

There are cases, when the iteration will not converge within the given knot intervals. Thus, iteration is cancelled if one of following criteria is met:

- The iteration diverges after converging.
- u or v lie outside the parameter domain of the segment.
- The number of iterations exceeds a pre-defined maximum.

In our application we observed that the algorithm usually converges after three to four iterations. We set the maximum number of iterations to ten, to ensure convergence for almost all cases.

The implementation of this surface intersection algorithm is based on the OpenNURBS library [17].

VI. Results

A. B-spline refinement factor

The chosen value C in the surface refinement process (2) influences the size of the parametric intervals and therefore the number of SBBs. This affects how well Newton's method is converging. If it is chosen too large performance decreases, if it is chosen too small reliable convergence cannot be guaranteed. In practice a value between 30 and 40 provides good results, meaning that there were no performance issues and high convergence rates.

B. Performance analysis

There are certain requirements to achieve interactive feedback of the scan simulation. First, the frame rate is fixed to 60 frames per second (FPS). This is achieved by using multi-threading where the main thread is responsible for rendering only. All remaining threads perform ray-casting jobs. Those are available for each new frame with the current orientation of the scanner. To avoid an overflow of ray-casting jobs a thread will only start a new job after finishing its previous job. Therefore, we set a target number of rays per frame. If a job is not finished within $1/FPS$ seconds, additional jobs may be discarded. This technique yields both the maximum throughput and maximum interactivity. The application was tested on two systems one with a i7-4770k processor (Machine 1) and one with a Core2Quad Q9300 processor (Machine 2).

Target rays per frame	Machine 1	Machine 2
1000 Rays	∼60'000 Rays/s	∼60'000 Rays/s
3000 Rays	∼180'000 Rays/s	∼90'000 Rays/s
10000 Rays	∼415'000 Rays/s	∼145'00 Rays/s

TABLE I: B-spline surface results with $C = 30$.

Different targets of rays per frame have been evaluated to discover the upper bound of possible ray-surface intersections

Target rays per frame	Machine 1	Machine 2
3000 Rays	∼180'000 Rays/s	∼180'000 Rays/s
4000 Rays	∼240'000 Rays/s	∼190'000 Rays/s
10000 Rays	∼600'000 Rays/s	∼190'000 Rays/s

TABLE II: Triangle mesh results.

per second for each system. The results are presented in Tables I and II. In both cases, either triangle meshes or b-spline surfaces the processing speed is high enough to produce accurate point clouds of the scanned geometry. It is beneficial to keep the number of rays per second small in order to avoid a rapidly increasing size of generated points.

C. Point clouds

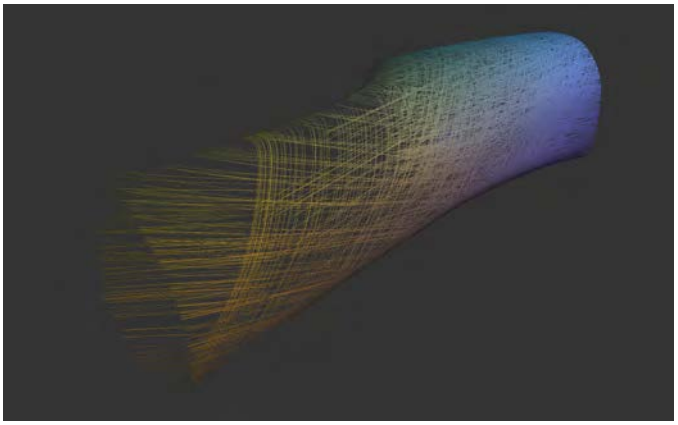


Fig. 8: Wishbone point cloud of a scanned B-Spline surface. The color scheme is created by interpolating colors over the scanning object and only serves a comfortable visual perception.

Each resulting point cloud has a unique scan structure. Each operator has a different scanning approach and generates a personal point cloud structure. Figure 8 shows a synthetic point cloud from a b-spline surface representing a wishbone model scanned with the presented system. On the left half of the figure the different angled scan lines are clearly visible. The right half of the point cloud has higher density, but the human factor is still visible. Figure 9 shows a point cloud result of scanning a triangular mesh rocker arm model. In the upper half of the figure, a cylinder can be seen. Especially on curved geometry with cutouts the human scanning approach cannot be predicted and will generate unique point clouds for each individual scan.

VII. Conclusion

In this paper we presented a virtual-reality 3d-Laser-Scan Simulation. Our solution makes scanning of CAD models inside a VR environment possible. Interaction with our system matches the process of operating a real hand-held laser scanner. The resulting point clouds cannot be differentiated from non-synthetic scans. Interactivity and execution speed of our application meet the expectations. The scan process

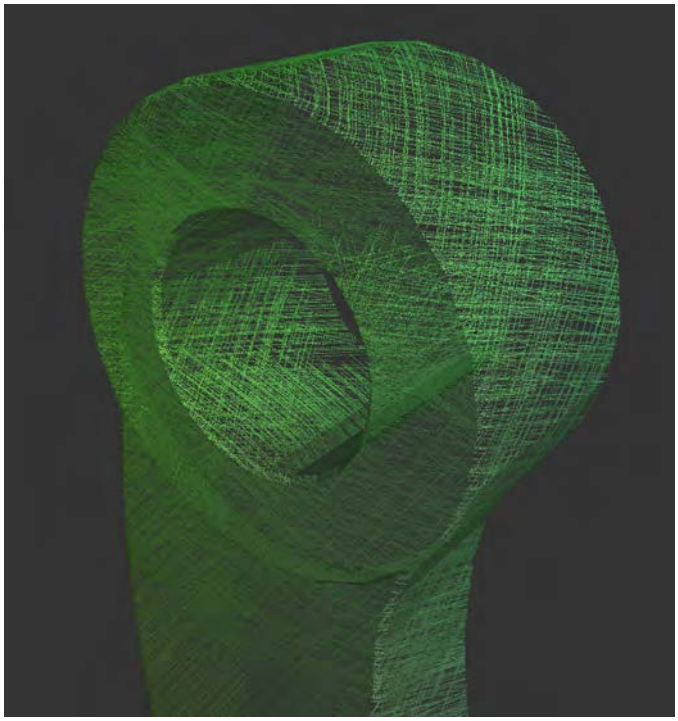


Fig. 9: Rocker arm point cloud of a scanned mesh.

is realistic and easy to perform. Nevertheless there are a few drawbacks. The Razer Hydra is not connected to a leading arm like some real laser scanners which leads to difficulties moving the scanner calm and precise. This could be compensated by an adjustable factor which decreases the sensitivity of the Hydra. Since the Oculus Rift is still in development there are a few points which will certainly improve with later versions. Especially the resolution causes the problem of a visible pixel grid and a blurry perception of concrete shapes on looking around in a scene. This problem will hopefully get fixed with newer versions which should have a higher pixel density.

References

[1] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg *et al.*, "The digital michelangelo project: 3d scanning of large statues," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 131–144.

[2] C. Y. Ip and S. K. Gupta, "Retrieving matching cad models by using partial 3d point clouds," *Computer-Aided Design and Applications*, vol. 4, no. 5, pp. 629–638, 2007.

[3] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas, "Registration of point cloud data from a geometric optimization perspective," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. ACM, 2004, pp. 22–31.

[4] F. Bernardini, C. L. Bajaj, J. Chen, and D. R. Schikore, "Automatic reconstruction of 3d cad models from digital scans," *International Journal of Computational Geometry & Applications*, vol. 9, no. 04n05, pp. 327–369, 1999.

[5] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, "Curve skeleton extraction from incomplete point cloud," in *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3. ACM, 2009, p. 71.

[6] M. Caputo, K. Denker, M. O. Franz, P. Laube, and G. Umlauf, "Support Vector Machines for Classification of Geometric Primitives in Point Clouds," in *Curves and Surfaces, 8th International Conference, Paris*

2014, J.-D. Boissonnat, A. Cohen, O. Gibaru, C. Gout, T. Lyche, M.-L. Mazure, and L. L. Schumaker, Eds., Springer. Springer, 2015, pp. 80–95.

[7] N. E. Seymour, A. G. Gallagher, S. A. Roman, M. K. OBrien, V. K. Bansal, D. K. Andersen, and R. M. Satava, "Virtual reality training improves operating room performance: results of a randomized, double-blinded study," *Annals of surgery*, vol. 236, no. 4, p. 458, 2002.

[8] H. G. Hoffman, W. J. Meyer III, M. Ramirez, L. Roberts, E. J. Seibel, B. Atzori, S. R. Sharar, and D. R. Patterson, "Feasibility of articulated arm mounted oculus rift virtual reality goggles for adjunctive pain control during occupational therapy in pediatric burn patients," *Cyberpsychology, Behavior, and Social Networking*, vol. 17, no. 6, pp. 397–401, 2014.

[9] A. Mossel, C. Schönauer, G. Gerstweiler, and H. Kaufmann, "Artifice-augmented reality framework for distributed collaboration," *International Journal of Virtual Reality*, 2013.

[10] "Oculus Runtime for Windows," https://developer.oculus.com/downloads/pc/0.5.0.1-beta/Oculus_SDK_for_Windows/, [09/03/2015].

[11] "Sixense Core API," http://sixense.com/sixensecoreapi, [09/20/2015].

[12] T. Akenine-Möller, "Fast 3d triangle-box overlap testing," in *ACM SIGGRAPH 2005 Courses*, ser. SIGGRAPH '05. ACM, 2005.

[13] T. Akenine-Möller, E. Haines, and N. Hoffman, *Real-Time Rendering Third Edition*. A K Peters, Ltd., 2008.

[14] D. Sunday, "Intersections of rays, segments, planes and triangles in 3d." http://geomalgorithms.com/a06-_intersect-2.html, [09/23/2015].

[15] "Barycentric Coordinate Computation by Dan Sunday." http://geomalgorithms.com/a04-_planes.html#Barycentric-Coordinate-Compute, [09/23/2015].

[16] W. Martin, E. Cohen, R. Fish, and P. Shirley, "Practical ray tracing of trimmed nurbs surfaces," *Journal of Graphics Tools Volume 5 Issue 1*, pp. 27–52, 2000.

[17] "OpenNurbs SDK," https://www.rhino3d.com/de/opennurbs, [09/07/2015].